DEVELOPMENT OF AN ADVANCED INTEGRATED CIRCUIT MEMORY TESTER

Approved for public release; distribution is unlimited.

July 2000

Prepared for:
Defense Threat Reduction Agency
45045 Aviation Drive
Dulles, VA 20166-7517

DNA001-92-C-0152

H. Jake Tausch
David Sleeter

Prepared by: Mission Research Corporation
5001 Indian School Road, NE
Albuquerque, NM 87110-3946

DTIC QUALITY INSPECTED 4

Technical Report

20001208 043

DESTRUCTION NOTICE:

DISTRIBUTION LIST UPDATE

This mailer is provided to enable DTRA to maintain current distribution lists for reports. (We would appreciate you providing the requested information.)

❑ Add the individual listed to your distribution list.

❑ Delete the cited organization/individual.

❑ Change of address.

> **Note:**
> Please return the mailing label from the document so that any additions, changes, corrections or deletions can be made easily. For distribution cancellation or more information call DTRA/ADM (703) 325-1036.

NAME: _____

ORGANIZATION: _____

OLD ADDRESS	NEW ADDRESS
_____	_____
_____	_____
_____	_____

TELEPHONE NUMBER: (____) _____

| DTRA PUBLICATION NUMBER/TITLE | CHANGES/DELETIONS/ADDITIONS, etc.) |
	(Attach Sheet if more Space is Required)
_____	_____
_____	_____
_____	_____

DTRA or other GOVERNMENT CONTRACT NUMBER: _____

CERTIFICATION of NEED-TO-KNOW BY GOVERNMENT SPONSOR (if other than DTRA):

SPONSORING ORGANIZATION: _____

CONTRACTING OFFICER or REPRESENTATIVE: _____

SIGNATURE: _____

CUT HERE AND RETURN

DEFENSE THREAT REDUCTION AGENCY
ATTN: ADM
45045 AVIATION DRIVE
DULLES, VA 20166-7517

DEFENSE THREAT REDUCTION AGENCY
ATTN: ADM
6801 TELEGRAPH ROAD
ALEXANDRIA, VA 22310-3398

REPORT DOCUMENTATION PAGE

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	JULY 2000	Technical 920904 - 960729

4. TITLE AND SUBTITLE

Development of an Advanced Integrated Circuit Memory Tester

5. FUNDING NUMBERS

C-DNA 001-92-C-0152
PE - 62715H
PR - AF
TA - DE
WU - DH00030

6. AUTHOR(S)

H. Jake Tausch and David Sleeter

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Mission Research Corporation
5001 Indian School Road NE
Albuquerque, NM 87110-3946

8. PERFORMING ORGANIZATION REPORT NUMBER

MRC/ABQ-R-1805 Revised

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Defense Threat Reduction Agency
45045 Aviation Drive
Dulles, VA 20166-7517
NSSE/Cohn

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

DSWA-TR-96-59

11. SUPPLEMENTARY NOTES

This work was sponsored by the Defense Threat Reduction Agency under RDT&E RMC code B 4662 D AF DE 00030 7010 A 25904D.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

This program developed a high speed portable integrated circuit tester which can generate sophisticated memory patterns and is capable of performing *in-situ* diagnostics to facilitate the identification of circuit failures.

14. SUBJECT TERMS

Integrated Circuit   Memory
Testing   Radiation Testing
ASIC

15. NUMBER OF PAGES

63

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAR

SUMMARY

The purpose of this program was to develop and demonstrate a portable integrated circuit tester to support radiation testing (e.g. SEU, dose-rate upset, and survivability, total dose testing, etc.). Radiation testing is an integral part of the development of high performance radiation hardened microelectronics. Prior to this program there existed no portable integrated circuit tester which provided the required excitation and was capable of performing in-situ diagnostics at radiation facilities at functional speeds designed for the test devices.

This program developed a high speed integrated circuit memory tester which is readily portable, can generate sophisticated memory patterns, and is capable of performing in_situ diagnostics to facilitate the identification of circuit failures. This tester is called the Algorithmic/Test Vector^{TM} (ATV^{TM}) system.

The compact size and high performance of the tester was largely achieved through the use of a full custom ASIC (Application Specific Integrated Circuit). This ASIC provides 8 channels of high speed formatting, delay and co-axial drive and replaces the equivalent of 24 integrated circuits.

The design has a simple, user friendly graphical interface which allows quick and easy programming of the system.

Effectiveness of the tester was demonstrated in May of 1996 when it was used to support heavy ion testing at the Brookhaven National Laboratory for the Radiation Tolerant Microelectronics program (DNA001-95-C-0041). In this program National Semiconductor Corporation produced CLAy31 configurable logic arrays which then needed to be characterized for SEU. Testing of the CLAy31s involved downloading a very complex programming file to the parts, periodically checking them for upsets, then re-programming the parts and logging errors when upsets were detected. The test was completely successful and demonstrated the effectiveness of the ATV tester.

ATV systems will provide advanced testing and diagnostic capabilities to users in the radiation effects community and will also be a valuable alternative to commercial firms that produce or test memories, ASICs, CPLDs, or any logic device. We have developed a detailed plan for marketing and selling these systems. Key aspects of the plan include minimizing maintenance support requirements by simplifying user interactions, extensive on-line help capability, and repair of systems through board replacement.

CONVERSION TABLE

Conversion factors for U.S. Customary to metric (SI) units of measurement.

MULTIPLY ⟶ BY ⟶ TO GET

TO GET ⟵ BY ⟵ DIVIDE

angstrom	$1.000\,000 \times$ E -10	meters (m)
atmosphere (normal)	$1.013\,25 \times$ E +2	kilopascal (kPa)
bar	$1.000\,000 \times$ E +2	kilopascal (kPa)
barn	$1.000\,000 \times$ E -28	meter^2 (m^2)
British thermal unit (thermochemical)	$1.054\,350 \times$ E +3	joule (J)
calorie (thermochemical)	$4.184\,000$	joule (J)
cal (thermochemical/cm^2)	$4.184\,000 \times$ E -2	megajoule/m^2 (MJ/m^2)
curie	$3.700\,000 \times$ E +1	*gigabecquerel (GBq)
degree (angle)	$1.745\,329 \times$ E -2	radian (rad)
degree Fahrenheit	$t_k = (t_f + 459.67)/1.8$	kelvin (K)
electron volt	$1.602\,19 \times$ E -19	joule (J)
erg	$1.000\,000 \times$ E -7	joule (J)
erg/second	$1.000\,000 \times$ E -7	watt (W)
foot	$3.048\,000 \times$ E -1	meter (m)
foot-pound-force	$1.355\,818$	joule (J)
gallon (U.S. liquid)	$3.785\,412 \times$ E -3	meter^3 (m^3)
inch	$2.540\,000 \times$ E -2	meter (m)
jerk	$1.000\,000 \times$ E +9	joule (J)
joule/kilogram (J/kg) radiation dose absorbed	$1.000\,000$	Gray (Gy)
kilotons	4.183	terajoules
kip (1000 lbf)	$4.448\,222 \times$ E +3	newton (N)
kip/inch^2 (ksi)	$6.894\,757 \times$ E +3	kilopascal (kPa)
ktap	$1.000\,000 \times$ E +2	newton-second/m^2 (N·s/m^2)
micron	$1.000\,000 \times$ E -6	meter (m)
mil	$2.540\,000 \times$ E -5	meter (m)
mile (international)	$1.609\,344 \times$ E +3	meter (m)
ounce	$2.834\,952 \times$ E -2	kilogram (kg)
pound-force (lbs avoirdupois)	$4.448\,222$	newton (N)
pound-force inch	$1.129\,848 \times$ E -1	newton-meter (N·m)
pound-force/inch	$1.751\,268 \times$ E +2	newton/meter (N/m)
pound-force/foot^2	$4.788\,026 \times$ E -2	kilopascal (kPa)
pound-force/inch^2 (psi)	$6.894\,757$	kilopascal (kPa)
pound-mass (lbm avoirdupois)	$4.535\,924 \times$ E -1	kilogram (kg)
pound-mass-foot^2 (moment of inertia)	$4.214\,011 \times$ E -2	kilogram-meter^2 (kg·m^2)
pound-mass/foot^3	$1.601\,846 \times$ E +1	kilogram/meter^3 (kg/m^3)
rad (radiation dose absorbed)	$1.000\,000 \times$ E -2	**Gray (Gy)
roentgen	$2.579\,760 \times$ E -4	coulomb/kilogram (C/kg)
shake	$1.000\,000 \times$ E -8	second (s)
slug	$1.459\,390 \times$ E +1	kilogram (kg)
torr (mm Hg, 0° C)	$1.333\,22 \times$ E -1	kilopascal (kPa)

*The becquerel (Bq) is the SI unit of radioactivity; 1 Bq = 1 event/s.
**The Gray (Gy) is the SI unit of absorbed radiation.

TABLE OF CONTENTS

FIGURES

SECTION 1
SYSTEM OVERVIEW

The Algorithmic/Test Vector (ATV) system is a low cost, portable, high performance digital tester which represents a major advance and simplification in the art of integrated circuit testing. Its modular architecture and custom ASICs achieve dramatic miniaturization and flexibility while delivering performance found in console systems costing many times as much. A picture of the ATV system is shown in Figure 1-1 and a functional diagram of its operation is shown in Figure 1-2.

1.1   SIMPLE TEST DEVELOPMENT.

With ATV, test development is broken into three separate elements, organized the natural way engineers work with parts:

Part/Package Description
Timing Diagrams
Test Programs

Each of these elements are created and stored separately, then combined at run time. Dynamic combinations allow simple yet powerful test variations. A new part can be tested by merely creating a new package description. Speed enhancements for next generation parts only require new timing diagrams. A single test program can be used for an entire family of parts in various packages. Due to this modularity, dramatically fewer programming elements need to be maintained in your archive. Productivity is enhanced since new tests are assembled using existing pin/package, timing and test program building blocks.

Part/Package Description - Test programming begins by assigning names to individual pins or groups of pins. This is accomplished graphically on an illustration resembling the DUT/card.

Timing Diagrams - Each ATV pin has independent timing, and can store up to 7 timing sets which are selectable on-the-fly. Timing is specified by graphically creating timing diagrams similar to the ones found in part specification sheets. Waveforms in each timing diagram are matched to pin names from the part/package description to configure pins.

ATV formats include NRZ (Non-Return to Zero), Return to One, Return to Zero, or Return to Complement. Timing diagrams can be as short as 20ns and edges placed with 100 ps resolution.
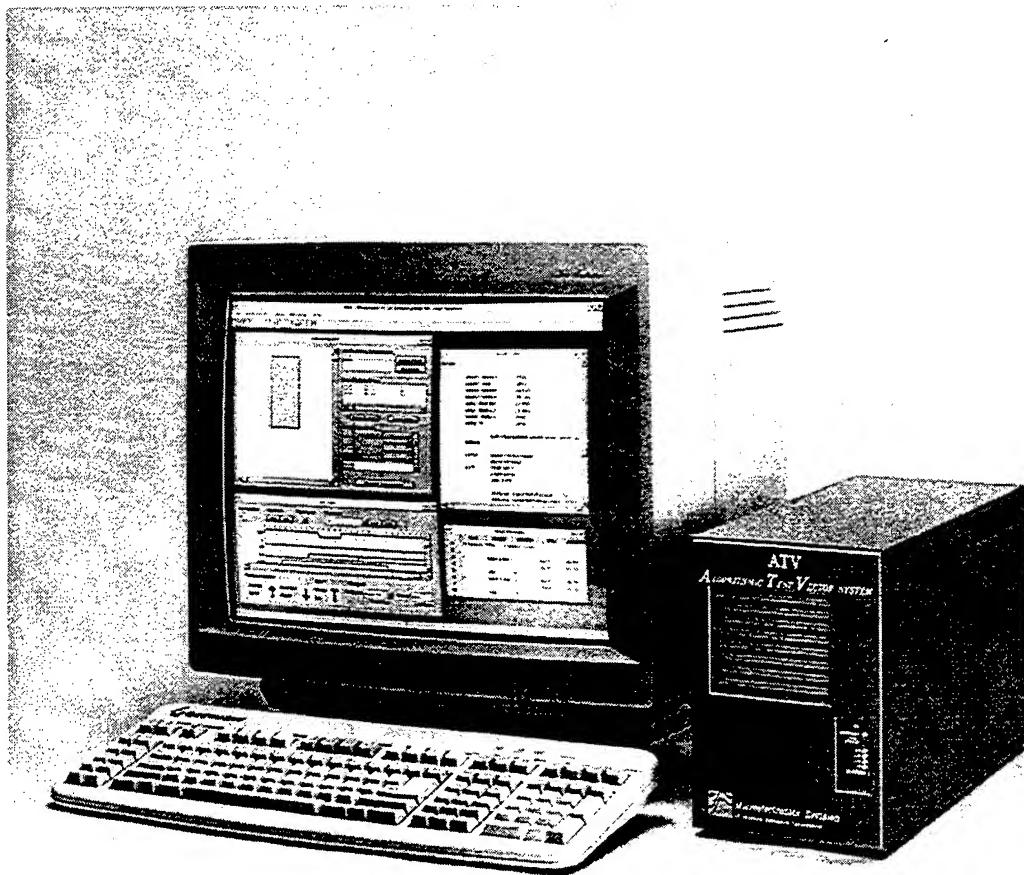
Figure 1-1. Photograph of the ATV system.
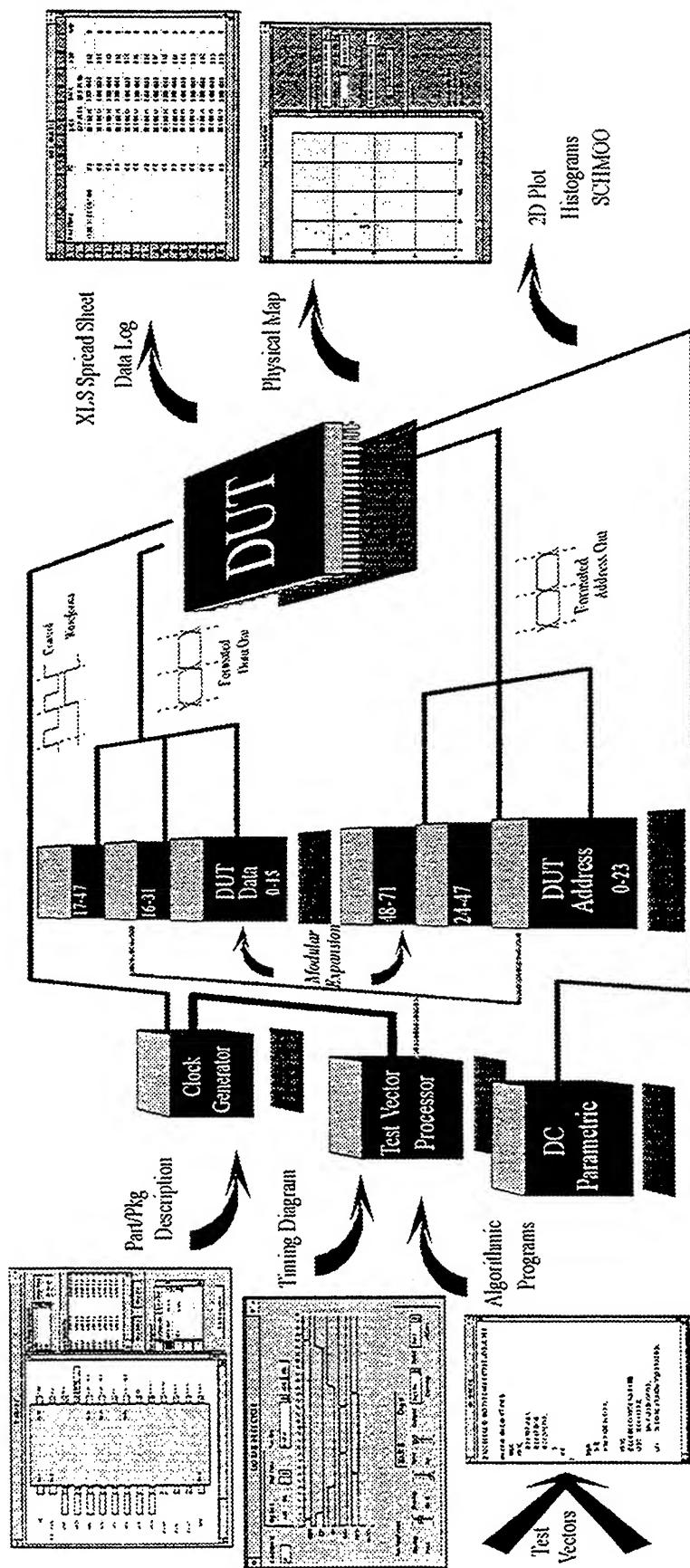
Figure 1-2. Functional overview of the ATV system.

3

Test Programs - ATV programs are developed in an integrated environment which compiles high level and assembly source code and downloads these programs to the ATV unit. The source language includes subroutines and interrupts. Procedure libraries are used so test programs can be simply built by including existing common code.

Data patterns can be generated real-time algorithmically or can be test vectors imported from ASCII files or EXCEL[1] format spread sheets.

Algorithmic pattern generation results in the fastest possible test speeds because there is no pause to down-load part of a large test pattern file. They also give greater insight into device operation since their program structure reflects the logical operation of the DUT (device under test). Algorithmic programs provide a powerful tool for investigating device failures.

Most device architectures allow algorithmically generated data patterns. Memories are well known examples in which millions of patterns can be generated from a few bytes of code.

1.2   FAST HARDWARE LOOP.

ATV provides a hardware feature which repeats a section of code without the time penalty normally associated with software loops. Thus, ATV can algorithmically generate 50 million timing diagrams per second using loop constructs.

1.3   EXPANDABLE ARCHITECTURE.

ATV uses a synchronized timing architecture and distributed processing so systems can be expanded without limit. A system is comprised of a Test Vector Processor (TVP) card, clock generator, and some mixture of sixteen channel I/O, and twenty four channel Address/Output cards.

TVP Card - The TVP card controls algorithmic program flow for the entire system. It also has two programmable power supplies, one of which measures IDDQ currents, 6 output strobe lines, trigger-in and trigger-out controls.

Input/Output (I/O) Cards - I/O cards have 16 input and output lines which can be used as I/O pairs or separately. Output patterns can be algorithmically generated or imported from test vector files.

Address/Output Cards - Address/Output cards have 24 channels. Output patterns can be algorithmically generated or imported from test vector files. Alternately, some or all of the output channels can be assigned to an address counter which can be initialized and then incremented/decremented. The address counter has an arithmetic logic unit which can generate complex address sequences on-the-fly. Output pins can be multiplexed for testing segmented address devices such as multiplexed row-address/column-address (RAS/CAS) on DRAMs.

---

[1] EXCEL is a trademark of Microsoft.

1.4   VIRTUAL TEST HEAD^{TM}.

ATV introduces the 'Virtual Test Head' which does away with the large, active electronic test heads and associated mechanical manipulators prevalent in most test environments.

In the ATV design all active electronics are kept in the controller and connections to the DUT are made by a simple cable set. Even without active test head electronics, ATV architecture provides programmable loads for the part. Configuring for bench top testing, wafer probing or part handling is as simple as plugging in the cable set.

Programmable Thevinin Load - ATV I/O channels have three programmable drive conditions, all of which are asserted on-the-fly as specified in the timing diagrams. Logic high and logic low are used when sending data to the DUT. The third drive condition, reverse terminate, is used to develop a programmable Thevinin load when the DUT is out-putting a signal. A Thevinin load consists of a series resistance back terminated in a voltage source. The reverse terminate drive establishes the voltage source for the load and resistors on the DUT card provide the series resistance.

1.5   DATA LOG AND ANALYSIS TOOLS.

ATV test data is logged in EXCEL format spread sheets. A built in spread sheet engine allows for easy inspection, manipulation, and post processing of data.

ATV includes an error mapping tool which displays a logical location of errors or their physical location on a die. It also has a chart tool for generating 2D plots, histograms and SCHMOO plots.

SECTION 2
SPECIFICATIONS

Preliminary specifications for the ATV system are given below:

Test Capabilities
>   Algorithmic Test Vector Generation
>   Import Test Vectors
>   50 Million waveforms (timing diagrams) per second
>   Trig-Out and Trig-In to synchronize with external hardware
>   Programmable Loads

Pin Electronics
>   Independent Pin Timing
>   Three Programmable Drive Levels: High Drive, Low Drive, Reverse Terminate ... all
>   Applied at Test Speeds
>   7 Timing Sets per Pin, Selectable On-The-Fly
>   Drive Formats: RTRN0, RTRN1, RTRNC, NRZ with Edge Placement of 125pS ...
>   Reverse Terminate Drive Applied During DUT outputs
>   Minimum Cycle Time 20nS (50MhZ) per Complete Waveform
>   10%-90% Rise Time of 2nS (5V signal, 6 feet of cable)
>   Formatted output voltage range:0 to 7V
>   Outputs Tri-state-able
>   Pins Switchable to External Connector for Parametric Testing

TVP Card
>   Six High Speed Formatted Output Pins ... Two Outputs Programmable 0 to 14V for DUT
>   Programming
>   Two Programmable Power Supplies: 0 to +7 volts, 1 amp total current
>   IDDQ Measurements: 5nA to 500mA in Four Ranges
>   Trigger In/ Trigger Out

Data Card
>   16 High Speed Formatted Output Pins
>   16 Input Pins with Dual Level Comparators: Valid Low and Valid High

Address/Output-Only Card
>   24 High Speed Formatted Output Pins
>   Address Generation On-The-Fly: Increment/Decrement/ALU
>   Size:(Varies with Number of Test Pins)
>   0.6 cu. ft. (Typical for 62 pin unit)   3.2 cu. ft. (Typical for 574 pin unit)
>   Weight: (Varies with Number of Test Pins)

SECTION 3
HARDWARE DESCRIPTION

An ATV system consists of a PC type controlling computer and one or more chassis containing ATV cards. There are four types of ATV cards: Test Vector Processor (TVP) Card, Hub Card, 16 channel I/O card, and 24 channel Output Only Card. Each system requires one TVP card and each chassis requires one Hub Card. Other chassis slots can be used for I/O and Output Only cards.

The ATV system was designed to achieve 50 MHz algorithmic flow with all system output signals synchronized to within several hundred pico-seconds, even when system cards are distributed across several chassis. This is achieved by separately controlling program flow and high speed timing.

Program flow is controlled by the Test Vector Processor (TVP) card which generates algorithm control information for other cards in the system. This information is distributed throughout the system via the hub cards. In the main chassis the TVP card is connected to the hub card and sends the algorithm control information to it. From there the data is distributed to hub cards in other chassis and from each hub card to the ATV cards in that particular chassis.

High speed timing is controlled within each chassis by a high speed, multi-phase clock located on each hub card. Clocks on the hub cards are phase-locked together so that timing accuracy is maintained across all chassis. Algorithmic information is combined with the high speed clocks in custom ASICs to generate the high speed output signals. These Format and Delay ASICs (FNDs) generate the formatted signals (e.g. Return To Complement), and provide individually adjustable delays for every pin/waveform.

Figure 3-1 is an overview of a typical system which shows how the various cards interact. Note in this figure that there is a controlling PC and two ATV chassis. The PC communicates with the ATV cards via expansion cards, one of which is located in the PC and one in each of the chassis. The PC uses this link to program ATV cards, to upload error patterns after a test, and to handshake with the TVP during a test.
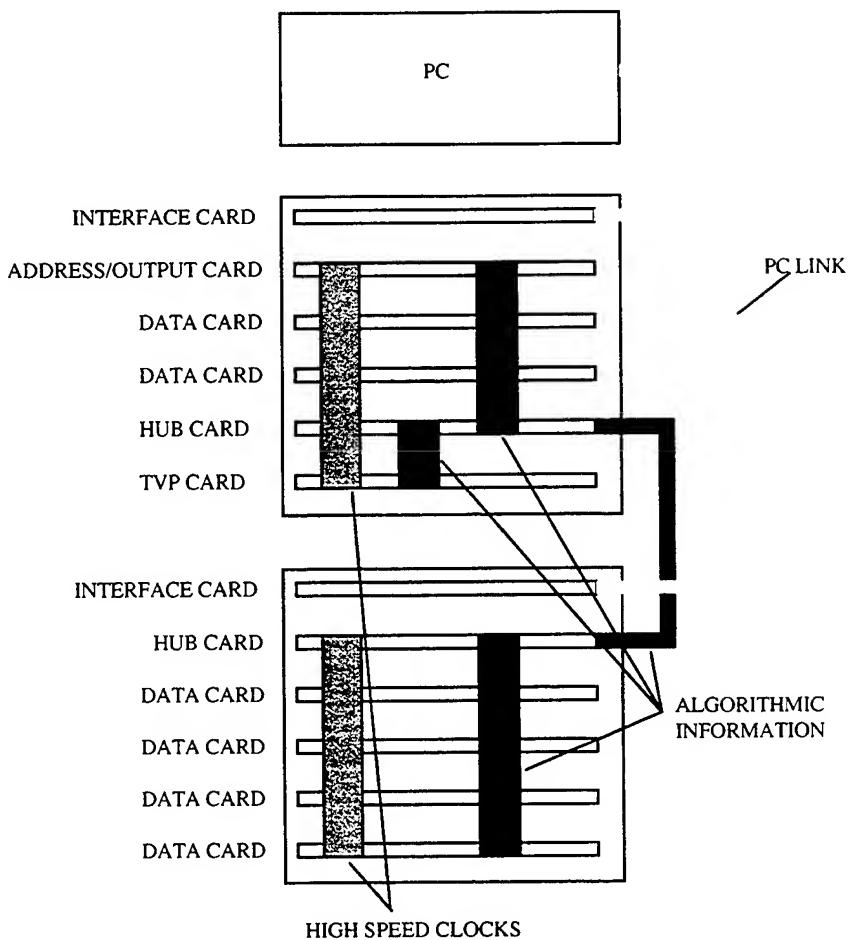
Figure 3-1. System interconnections.

Figure 3-2 is a functional block diagram of the TVP board. The main function of the TVP board is to control the algorithmic test flow of the system. The Test Vector Processor itself is a custom RISC processor that executes programs from on-board memory. The system PC downloads this memory and starts the TVP running at the beginning of a test. Once it is running, the TVP and PC can communicate with each other through an 8 bit mailbox to co-ordinate such things as start-up conditions, end of test, change bias, etc.

The TVP card has two separately programmable voltage sources with programmable current limits that can be used to bias up test devices. One of the voltage sources also has an IDDQ detection circuit.

The TVP card also has TTL level trigger-out and trigger-in signals that can control external circuitry or be used to co-ordinate algorithmic execution based on external events.

The TVP card also has 6 general purpose output strobes that can be used.
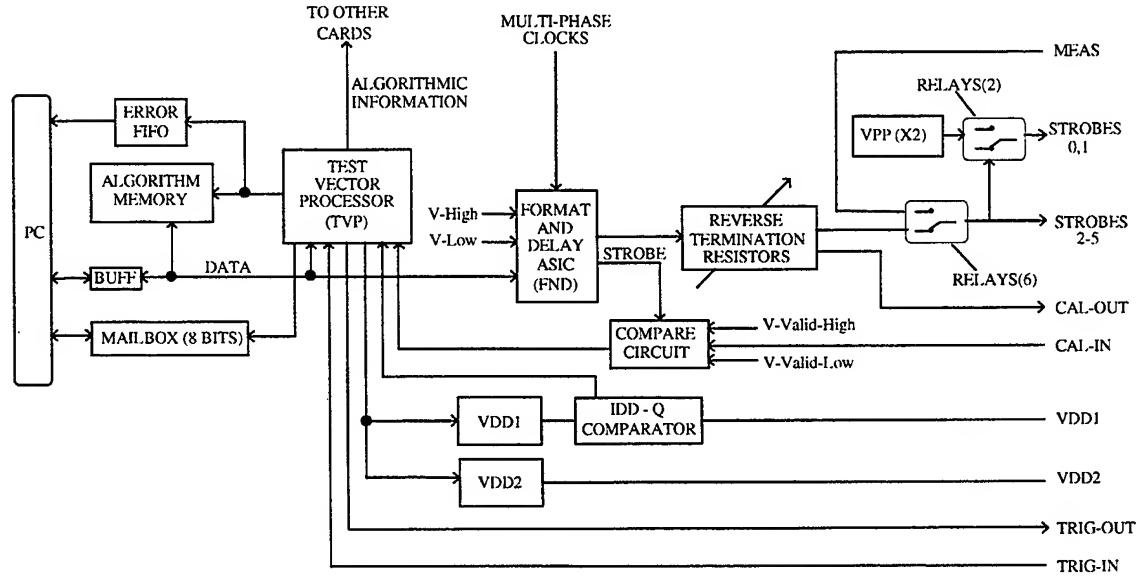
Figure 3-2. Functional block diagram of the TVP board.

Two of the strobe outputs can be connected through relays to programmable VPP sources for programming devices.

For separate stimulation or analog characterization, any combination of the strobe output pins can be connected through relays to an SMB connector (MEAS) on the rear of the card.

Two signals are provided over SMB connectors for high speed calibration. CAL-OUT is a 50 ohm reverse terminated strobe that provides stimulation and CAL-IN is a 90 ohm terminated input to a high speed comparator circuit.

The functional diagram of a data board is shown in Figure 3-3. Algorithmic information is used by this board to generate output data patterns and to compare data patterns against data received from the DUT. When an error is detected, the patterns of expected data and errors are stored in a FIFO and later read by the PC.

For separate stimulation or analog characterization, any combination of the data output pins can be connected through relays to an SMB connector (MEAS) on the rear of the card.

9
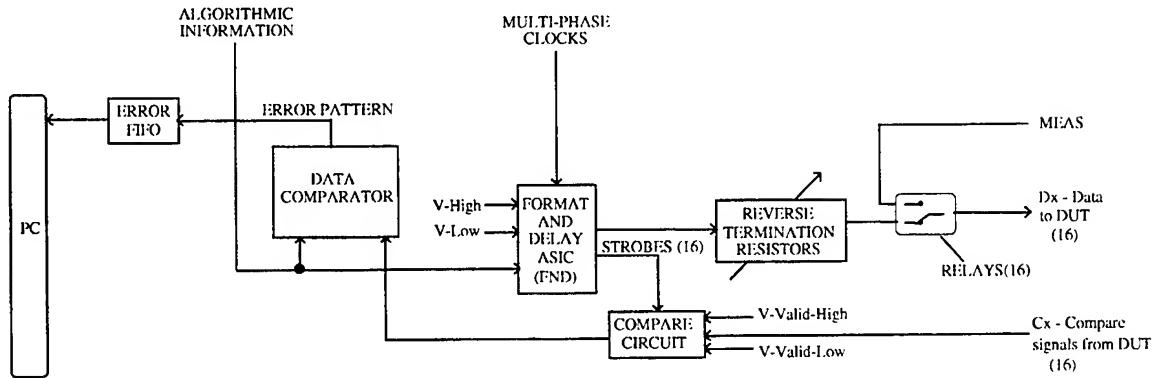
Figure 3-3. Functional block diagram of the data board.

Figure 3-4 is a functional diagram of an address/output only board. Algorithmic information is received by this card and used to generate data patterns output to the DUT. The address processor on this card can output patterns directly, or can output patterns generated by an internal up/down counter. The address processor also has an internal arithmetic logic unit which can perform mathematical operations on the pattern stored in the counter. The address processor also has the ability to multiplex high and low counter values onto the lower order pins for multiplexing RAS/CAS values in DRAMs.

For separate stimulation or analog characterization, any combination of the data output pins can be connected through relays to an SMB connector (MEAS) on the rear of the card.
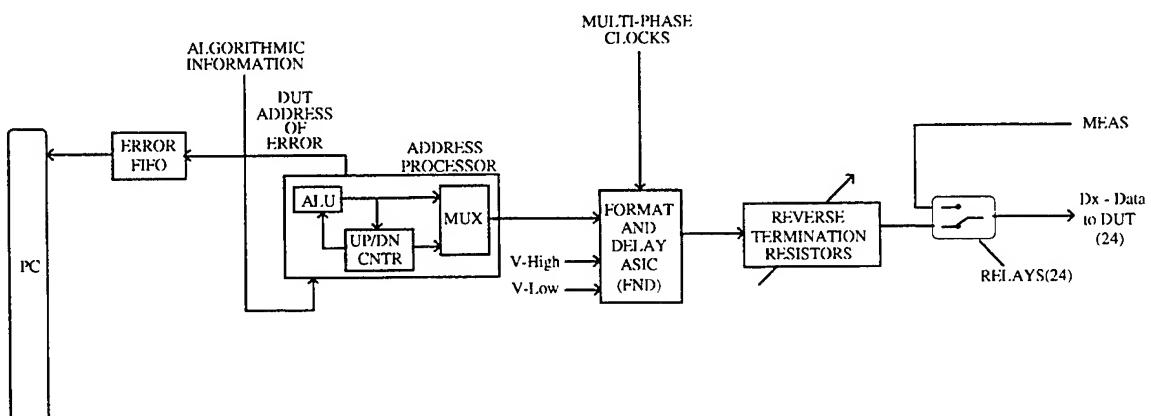

Figure 3-4. Functional block diagram of the address board.

10

3.1 ADJUSTMENTS.

There are three types of adjustments required for each card. The card's address must be set using dip switch selectors, each cards reference voltage must be adjusted, and the reverse terminate impedance must be trimmed on each high-speed output pin.

Reference voltages are adjusted by connecting a DVM to the appropriate test point on each card and adjusting a trim pot to set the measured voltage to 9 volts.

Reverse terminate impedance can best be adjusted by observing output signals at the end of a coaxial cable and adjusting the appropriate trim pot.

The following figures provide details for adjusting each card type.

Figure 3-5 shows the physical location of calibration components on the TVP card and also shows the pinout of signals on the cards rear connector.

The card's reference voltage is adjusted by monitoring the voltage on TP12 while adjusting R109. This voltage should be set to 9 volts.

The reverse termination of each high speed output can be set by adjusting the trim resistor paired with each output on the figure. For instance, channel D0 can be adjusted using R107. On the figure the resistor designator is followed by the letter "b" indicating the resistor is physically located on the opposite side of the board from the one shown.
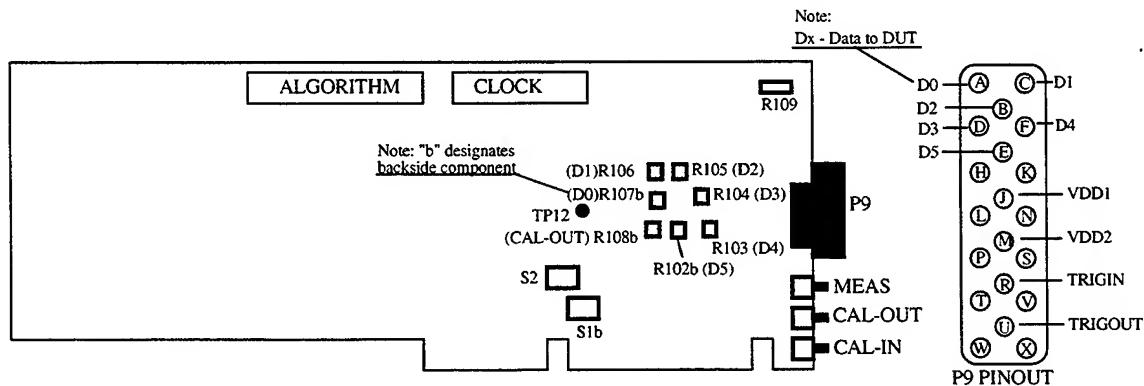
Figure 3-5. Physical overview of TVP board.

Figure 3-6 shows the physical location of calibration components on each data card and also shows the pinout of signals on the cards rear connector.

11

The card's reference voltage is adjusted by monitoring the voltage on TP11 while adjusting R15. This voltage should be set to 9 volts.

The reverse termination of each high speed output can be set by adjusting the trim resistor paired with each output on the figure. For instance, channel D0 can be adjusted using R31. Resistor designators on the figure that are followed by the letter "b" indicate the resistor is physically located on the opposite side of the board from the one shown.
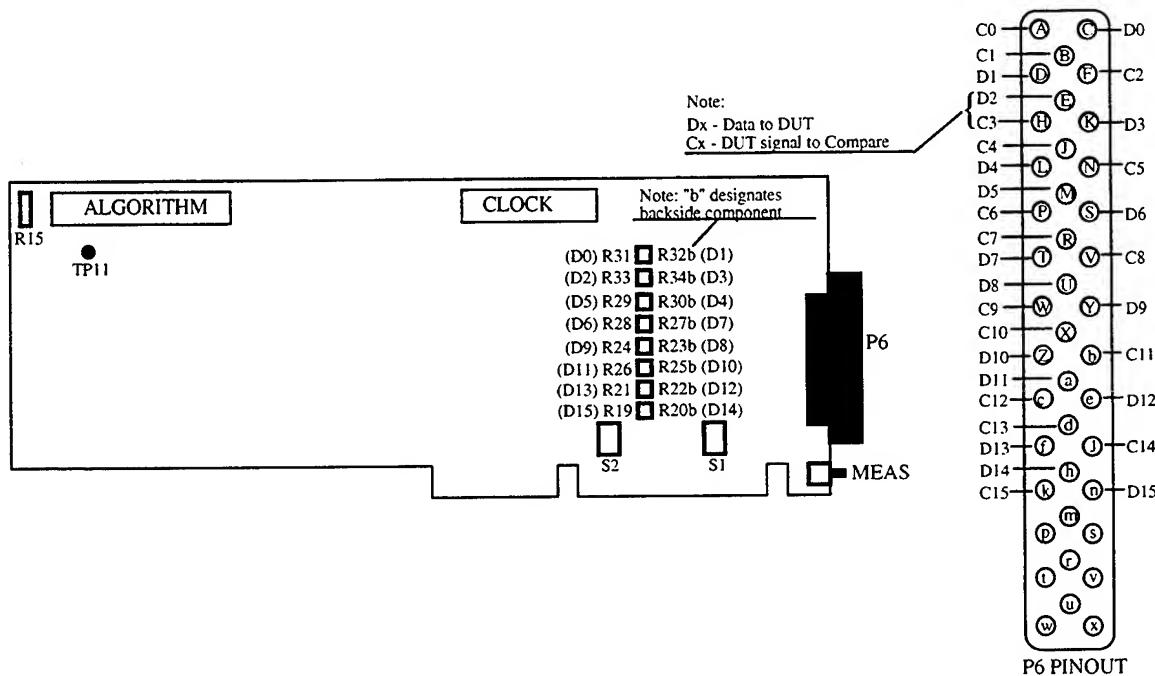
Figure 3-6. Physical overview of data board.

Figure 3-7 shows the physical location of calibration components on each data card and also shows the pinout of signals on the cards rear connector.

The card's reference voltage is adjusted by monitoring the voltage on TP9 while adjusting R37. This voltage should be set to 9 volts.

The reverse termination of each high speed output can be set by adjusting the trim resistor paired with each output on the figure. For instance, channel D0 can be adjusted using R53. Resistor designators on the figure that are followed by the letter "b" indicate the resistor is physically located on the opposite side of the board from the one shown.
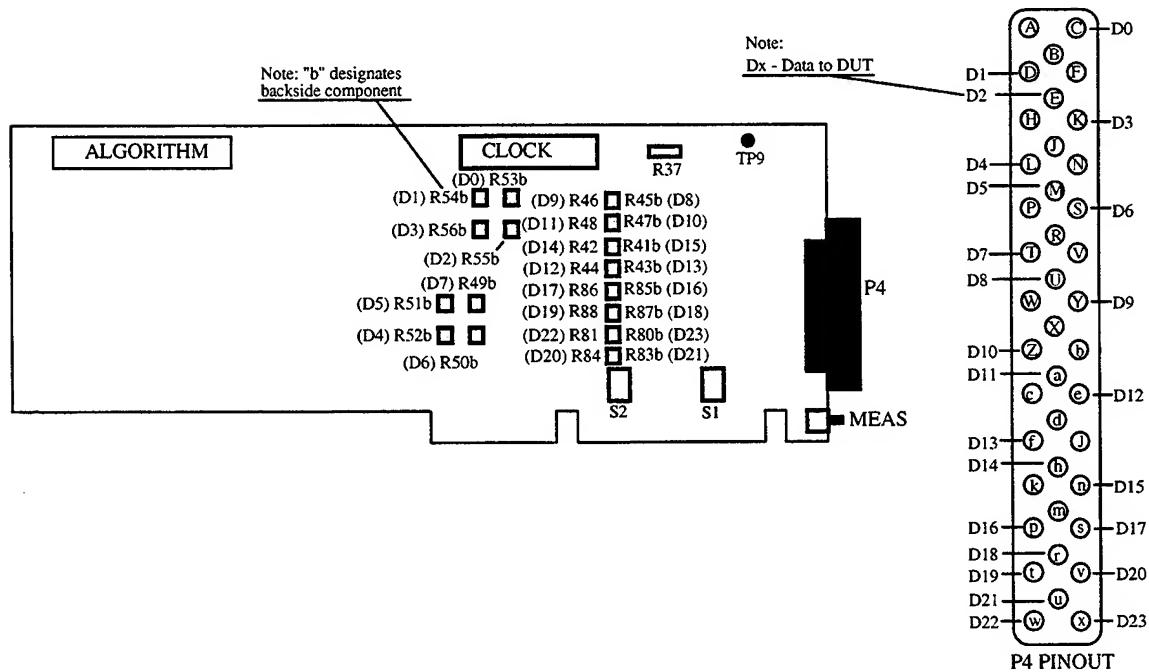
Figure 3-7. Physical overview of address board.

3.2 EXPANDING TO MULTIPLE CHASSIS.

In multiple chassis configurations the individual Hub Cards in each chassis will be interconnected with and phase-locked to each other so that timing will be maintained for all output pins. One Hub Card is designated the master by installing jumpers J3, J4, J5, J6 and J7. Other Hub cards will be configured as slaves by removing these jumpers. Interconnections are made by connecting a co-axial cable between the SMB connectors labeled P4 on the rear of the hub cards and by connecting a 60 pin ribbon cable between the ribbon connectors labeled P5 on the rear of the hub cards.

13

SECTION 4
PROGRAMMABLE THEVININ LOAD

A Thevinin load consists of a series resistance back terminated in a voltage source as shown in Figure 4-1.   Given any two current/voltage bias conditions, a Thevinin load can be calculated which will produce those conditions.

```
                                    V-Load1 @
                                    I-Load1

V-Thevinin  ——————/\/—————    AND
                R-Thevinin
                                    V-Load2 @
                                    I-Load2
```
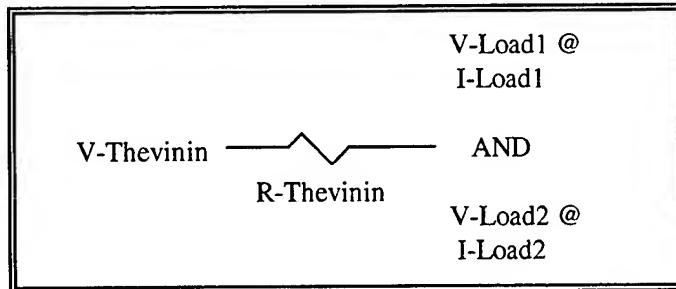
Figure 4-1. Thevinin load.

The Thevinin resistance and voltage for a given load can be calculated as follows.

$$\text{I-load1} * \text{R-Thevinin} = (\text{V-load1} - \text{V-Thevinin})$$
$$\text{I-load2} * \text{R-Thevinin} = (\text{V-load2} - \text{V-Thevinin})$$

Subtracting:

$$\text{R-Thevinin} = (\text{V-load1} - \text{V-load2})/(\text{I-load1} - \text{I-load2})$$

And:

$$\text{V-Thevinin} = \text{V-load1} - \text{I-load1} * \text{R-Thevinin}$$

A typical part might want to have a load of +5mA when its output was at 4.5V and a load of -15mA when its output was 0.5V. From the above equations it is easy to see that R-Thevinin should be 200 ohms and V-Thevinin should be 1.5V.

Each ATV I/O channel has two co-axial connections. A back terminated 50 ohm driver outputs voltages to provide logic high and logic low signals to the DUT and also to provide a third voltage for the Thevinin load when the DUT is outputting a signal. Signals are returned from the DUT over a separate 90 ohm coax cable which is resistively terminated. These co-axial signals

are typically tied together at the DUT card as shown in Figure 4-2. This figure also shows how a resistor can be added to the DUT card to complete the Thevinin load.

The resistive load as seen by the DUT will be the resistor on the DUT card plus the parallel combination of 50 and 90 ohms (32 ohms). In the above example where R-Thevinin was 200 ohms, a resistor of 168 ohms would be placed on the DUT card in series with each output. The equivalent Thevinin voltage as seen by the DUT will be the voltage from the 50 ohm driver times 0.643 which is the voltage divider ratio of the 50 and 90 ohm circuits. In the above example the 50 ohm drive voltage would be 2.33V to produce a Thevinin voltage of 1.5V.
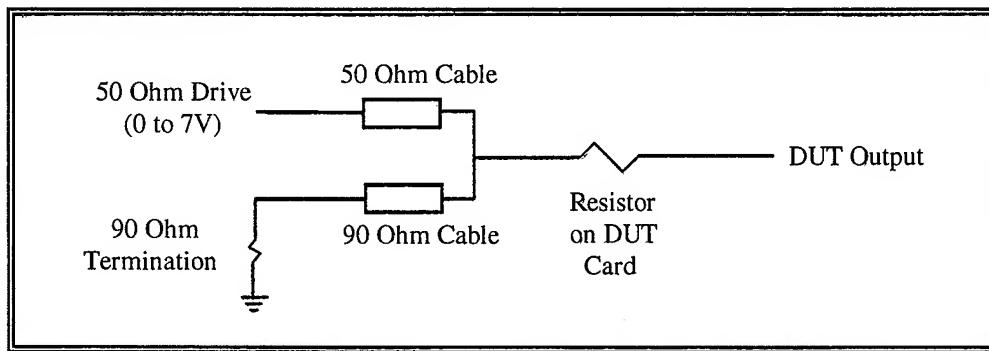
Figure 4-2. Thevinin termination circuit.

Note that placing a 168 ohm resistor in series with each DUT I/O pin should not have any significant effect when sending data to the DUT because the inputs of modern circuits typically have very high impedances. The resistor will affect the magnitude of the DUT signal returned to the I/O channel and this must be taken into account. From simple circuit analysis it can be seen that the voltage at the 90 ohm termination (input to the voltage comparators) will be:

V-Comparator = V-Thevinin + (32 / (32 + R-Dut)) * (V-Dut - V-Thevinin)

Again using the above example, assume the specification for valid high DUT signals was 4V and for valid low signals was 1V. The actual comparator levels would then be set to 1.9V and 1.42V, respectively. Thus, the actual voltage difference between valid high and low at the DUT was 3V whereas the programmed difference at the comparators was 0.48V.

SECTION 5
HOW TO RUN A TEST

An ATV test consists of three independently generated components: a test fixture with pin assignments, timing diagrams and an algorithmic program. Information from these components are combined based on Pin and PinGroup names as illustrated below in Figure 5-1. In this example a group of pins in a test fixture are named "DATA." A timing diagram named "WR" has a waveform named "DATA." A "TDG" line in the algorithm applies the bit pattern "110" to the pins in the "DATA" group using timing from the "DATA" waveform in the "WR" timing diagram.
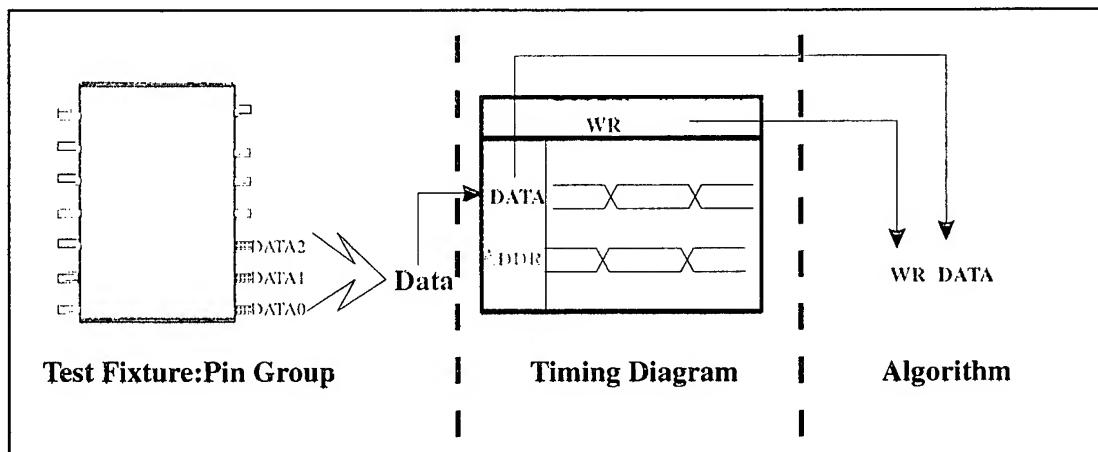
Figure 5-1. ATV combines test elements based on pin/pin group names.

Generally it is a good idea to plan naming conventions before creating any of these components. With proper planning you can combine sets of these components to create many different tests. The details for how to build test fixtures, timing diagrams and algorithms are given below. Once these components are generated a test is run as follows:

A test is created by linking together an algorithm with a text fixture and some number of timing diagrams in an Algorithm Project user interface. Algorithm Projects are accessed by selecting **Instruments|ATV|Algorithm**.

16

Before running a test, the algorithm must be compiled, and a down load file built and down-loaded to ATV. This is accomplished by selecting **Algorithm|Compile**, followed by selecting **Algorithm|Build Download**. Next download the test by selecting **Algorithm|Load.**

The test is then run by selecting **Algorithm|Run** at which point the Run Algorithm Dialog Box appears. Set the test frequency then run the test by pressing the Run button.

Test results are stored in a EXCEL style spread. Prior to running the test, Data Log parameters can be setup by selecting **Algorithm|DataLog Setup**. (Note: The spread sheet is initially displayed as an icon in the bottom left corner of the screen. To view test results while a test is running, double click on the spread sheet icon.)

5.1   TEST FIXTURES.

A Test Fixture is used to specify pin names for a Device Under Test (DUT). These pin names are used by loaded instrument drivers to stimulate the proper pins on the DUT. Test fixtures are designed by the user and added to a data base. The test fixture is a graphical element that visually portrays the actual part package, test board or interconnect of the DUT. The following sections describe how to create a test fixture, and how to make pin name assignments for a specific part in existing test fixtures.

5.1.1   Creating A Test Fixture.

Test fixtures are derived from **ASCII files** with a specific format. The ASCII files can be created externally and imported. Imported ASCII files can be created in a word processor, but are more typically created from a spread sheet. Spread sheet creation is useful and efficient when the placement of pins can be specified as spread sheet formulas. This method may be preferable when there are a large quantity of regularly spaced pins, or when several variations of a test fixture type are to be created. Example test fixture spread sheets are included in the **\examples** subdirectory.

Test Fixtures are composed of basic graphical objects including lines, filled and empty rectangles and ellipses, and text. These objects are used to build a visual image of a test fixture according to dimensions and aspects specified in the formatted ASCII file. The test fixture visually portrays the actual part package, test board or interconnect of the DUT. A sample test fixture is shown in Figure 5-2.

Graphic objects in the ASCII file can be one of two types: **Decorations** or **Hot Spots**. Decorations are inactive objects which the designer can use to help convey the visual look or titles of the test fixture. The picture is also built with definable '*hot spots*' that the user identifies as pins, and are selectable using the mouse. Hot spots are rectangle or ellipse objects that have a **Selectable** flag set to TRUE in the ASCII file. When the user selects a hot spot, it is highlighted along with two associated text objects. These text objects are pin name identifiers which the text fixture designer uses to cue the user to the function of the selectable pin. One of the text objects is replaced by the pin name that the user will specify when making pin assignments.

Section 5.1.2 describes details of the Test Fixture ASCII file format. This provides all the information necessary to construct test fixture files externally. Once a test fixture has been created it can be tested before loading into a data base. See Section 5.1.4, Testing And Loading A Text Fixture (*.fxt) File.
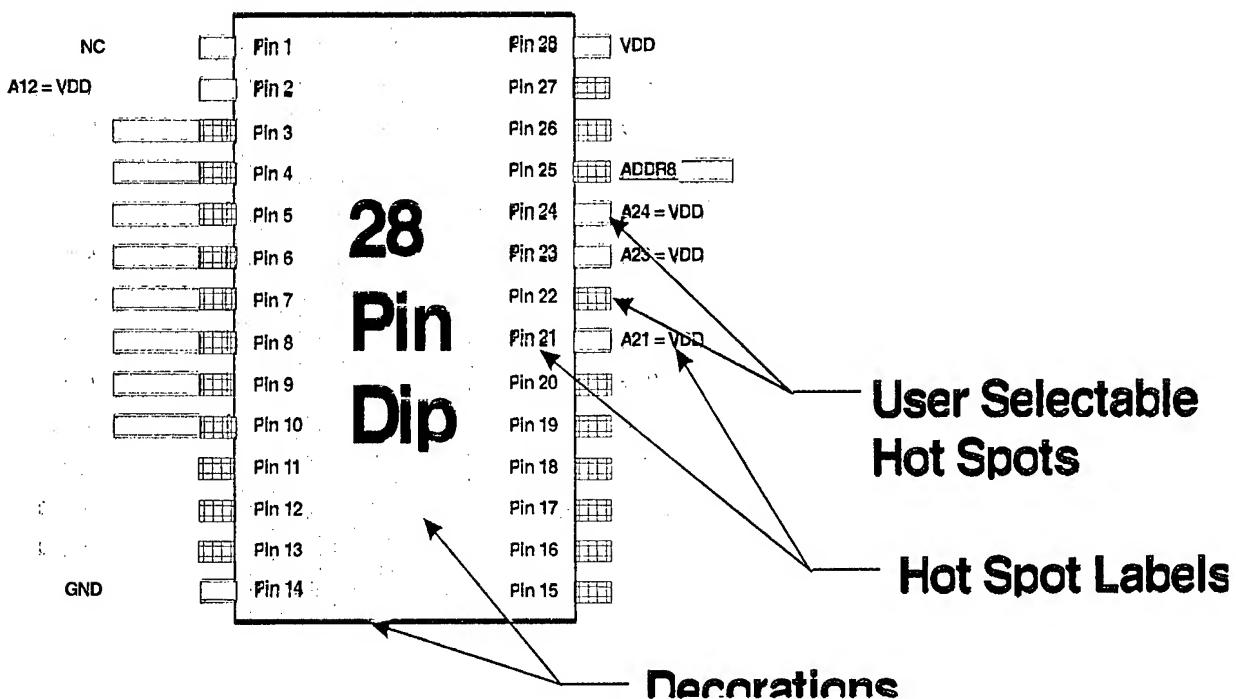
Figure 5-2. Sample test fixture.

5.1.2   Test Fixture ASCII File Format.

A sample excerpt from a test fixture ASCII file is given below in Figure 5-3:

The ASCII file must have a '.fxt' extension.

```
ATV-TEST-FIXTURE-DRAWING
14 Pin DIP
BEGIN-BLOCK      // Decoration Definitions
0
2
2  0  0  0  0  2  0  0  12632256  0  400  100  700  340
4  0  0  0  0  16777215  550  130  730  355  0  -900  10  14 Pin DIP
END-BLOCK          // End Definition of Decorations
BEGIN-BLOCK         // Pin Definitions for Connector 1
1
I/O 1
INPUT/OUTPUT
32
42
2  1  1  3  0  1  0  0  15780008  5  368  116  400  132
4  1  0  0  0  16777215  415  116  465  132  0  0  5  Pin 1
4  1  0  0  0  16711680  292  116  392  132  2  0  8  PIN (1)
2  2  1  3  0  1  0  0  15780008  5  368  148  400  164
4  2  0  0  0  16777215  415  148  465  164  0  0  5  Pin 2
4  2  0  0  0  16711680  292  148  392  164  2  0  8  PIN (2)


•  •  •  •
2  14  1  3  0  1  0  0  15780008  5  700  308  732  324
4  14  0  0  0  16777215  635  308  685  324  2  0  6  Pin 14
4  14  0  0  0  16711680  740  308  840  324  0  0  9  PIN (14)
END-BLOCK          // End Definition of Pins for Connector 1
```

Figure 5-3. Excerpt from a test fixture ASCII file.

Header - The string '**ATV-TEST-FIXTURE-DRAWING**' must appear at the top of the file followed by the test fixture name.

Blank Lines - Import error checking gives error messages with line numbers. Blank lines are not counted. If ASCII file includes blank lines, reported error line will not correlate to the file.

Comments must be preceded by two forward slashes '//'.

Blocks - Graphical objects in the ASCII file are arranged in groups or blocks. Blocks are initiated with the string '**BEGIN-BLOCK**' and end with the string '**END-BLOCK**'. There are two types of blocks defined as follows:

block type	*ascii code*
Decoration	0
Connector	1

There can be any number of blocks in any order.

Decoration blocks are groups of inactive graphical objects that are used to help convey the visual look or titles of the test fixture. The form of a Decoration block is as follows:

BEGIN-BLOCK

block type ascii code
number of graphical objects
{list of graphical objects: one object per line}
(See Graphical Object ASCII Formats Section 5.1.3)
END-BLOCK

Connector blocks define a group of pins which correspond to a test fixture connector. These pins are **hot spots** which the user of the test fixture can select. By convention each pin is defined by three graphical objects: A rectangle or ellipse followed by two text objects. All three of these objects are highlighted when the user of the test fixture clicks with the mouse to select the rectangle or ellipse. The first text object forms a title that is constant in the test fixture picture, the second text object forms a title which is replaced by a pin name assigned by the user. If the user de-assigns the pin, the second text object reverts to the default text in this ASCII file. The form of a Connector block is as follows:

BEGIN-BLOCK
block type ascii code
connector name
pin type
number of pins
number of graphical objects
{list of graphical objects: Sets of Three Objects-
(See Graphical Object ASCII Formats Section 5.1.3)
         rectangle/ellipse   ,
         fixed text label,
         replace-able text label}
END-BLOCK

The *connector name* is the text that will appear in the fixture connector column of the **Card Assignment** grid in the Open/Create Test Fixture Dialog Box. (See Part Pin Assignment Section 5.2). It cues the user to the type of connector defined for this block. By convention, all pins in a connector group are of the same *pin type* and must be one of the following:

OUTPUT-ONLY
INPUT/OUTPUT

20

The *number of pins* designates the total pins for this connector group. The pin type along with the number of pins control which Instrument Card types will appear in the list of cards in the system card column of the **Card Assignment** grid in the Open/Create Test Fixture Dialog Box. (See Part Pin Assignment Section 5.2). Specifically, only compatible system cards with the same pin type and same number of pins will be available to the user for assignment to this connector group.

The *number of graphical objects* will always be three times the number of pins.

5.1.3   Graphical Object ASCII Formats.

Graphical objects in a test fixture ASCII file are of the following types

object type	*ascii code*
Line	1
Rectangle	2
Ellipse	3
Text	4
Empty Rectangle	5
Empty Ellipse .	6

Following is an example of lines from a test fixture ASCII file showing the definition graphical objects in a Decoration Block and a Connector Block

```
2  0  0  0  0  2  0  0  12632256  0  400  100  700  340              (Rectangle Object in Decoration Block)
4  0  0  0  0  16777215  550  130  730  355  0  -900  10  14 Pin DIP  (Text Object in Connector Block)

2  1  1  3  -3  1  0  0  15780008  5  368  116  400  132             (Rectangle Object in Connector Block)
4  1  0  0  0  16777215  415  116  465  132  0  0  5  Pin 1          (Text Object in Connector Block)
4  1  0  0  0  16711680  292  116  392  132  2  0  8  PIN (1)        (Text Object in Connector Block)
```

Rectangle and **Ellipse** objects have the following format:
> *<object type> <card pin number> <selectable> <select type> <sibling>*
> *<pen width> <pen color>*
> *<brush style> <brush color> <brush hatch>*
> *<x1> <y1> <x2> <y2>*

Empty Rectangle and **Empty Ellipse** objects have the following format:
> *<object type> <card pin number> <selectable> <select type> <sibling>*
> *<pen width> <pen color>*
> *<x1> <y1> <x2> <y2>*

Text objects have the following format:

> <object type> <card pin number> <selectable> <select type> <sibling> <color>
> <left> <top> <right> <bottom>
> <text align> <rotate angle>
> <string length> <string>

The *object type* is as defined above.

The *card pin number* specifies which pin on the instrument card this connector pin is attached to. For Decoration Blocks, card pin number is 0.

The *selectable* parameter determines whether this object is a user selectable **hot spot**. (No=0, Yes = 1) For Decoration Blocks this parameter is 0. In a Connector Block, the rectangle/ellipse object of a pin set should be set to 1, the two associated text objects should be set to 0.

The *select type* parameter is an arbitrary number assigned to a pin set by the test fixture designer. It is used when the user is selecting groups of pins to assure that the pins are all of a compatible type. For Decoration Blocks this parameter is 0.

The *sibling* parameter is used by the test fixture designer to chain together some number of pins within a given connector. When a user selects a pin with siblings, the selected pin along with all of its siblings are highlighted. The typical use of the sibling parameter is to group pins into I/O pairs. Another example is for buss connections which contain multiple siblings. Siblings are specified as circular links. In other words, the last sibling in a set points back to the first. The number given in the sibling parameter specifies a relative line number for the next sibling in this sibling set. In the above example, the sibling parameter is '-3', indicating that the next sibling in this set is three lines back in the ASCII file.   For Decoration Blocks this parameter is 0. In a Connector Block, only the rectangle/ellipse object of a pin set may have a non-zero sibling number.

The *pen width* and *pen color* parameters control the pen style used to draw the object. The color is specified as a RGB four byte value defined as follows:

byte	meaning	values
high byte	na	0
byte 2	BLUE intensity	(0-255)
byte 1	GREEN intensity	(0-255)
low byte	Red intensity	(0-255)

The *brush color is defined the same way as pen color above. The brush style* and *brush hatch* parameters are used as follows:

brush styles	values
Solid	0
Hollow	1
Hatched	2
Pre-Defined Bit Maps	3..n

brush hatch	values
Horiz	0
Vert	1
LDiag	2
RDiag	3
Cross	4
DiagCross	5

The *x1*, *y1*, *x2* and *y2* parameters specify the top left and bottom right coordinates of the object in pixels. The origin of the coordinate system is the top left corner of the picture is (0,0), increasing x is to the left, increasing y is down.

The *color* parameter for text objects is defined the same way as pen color above.

The *left*, *top*, *right* and *bottom* parameters for text objects define a bounding rectangle for the text string. The text will be automatically sized to fit this rectangle.

The *text align* parameter is defined as follows:

byte	values
Left	0
Center	6
Right	2

The *rotate angle* parameter specifies a rotation angle for the text in 0.1 degree increments, counter clockwise from the three o'clock position. For example, a value of 450 will display text at 45 degrees.

The *string length* parameter specifies the number of characters in the *string* parameter including spaces.

5.1.4   Testing And Loading A Text Fixture (*.fxt) File.

After a Test Fixture ASCII files (*.fxt) is created, it must be loaded into a data base before it can be used.

5.1.5   Testing a Test Fixture ASCII File.

Prior to loading a test fixture, it is a good idea to error check the file. This is accomplished by selecting **File|Preview TstFxt File.** The ASCII file will be parsed and the graphic picture will be displayed. The preview halts upon errors found in the ASCII file. Errors are reported to the screen, giving the nature of the error and the line location. The test fixture picture will be drawn up to the place in the file where errors were detected.

5.1.6   Loading a Test Fixture ASCII File.

Loading a Test Fixture ASCII file into a data base is accomplished by selecting **File|New|TestFixture.** The Open File Dialog Box appears. Select the drive, path and file name of the (*.fxt) test fixture ASCII file. Select the data base that the test fixture will be loaded into. Press OK.

5.2   PART PIN ASSIGNMENT.

Pin assignments of a specific part in a test fixture must be made prior to running a test.
This is accomplished in a pin assignment Dialog Box which is accessed by selecting **File|New|TestFixture:PinConfig** or by selecting **File|Open|TestFixture:PinConfig**. The dialog box illustrated below in Figure 5-4 will appear.
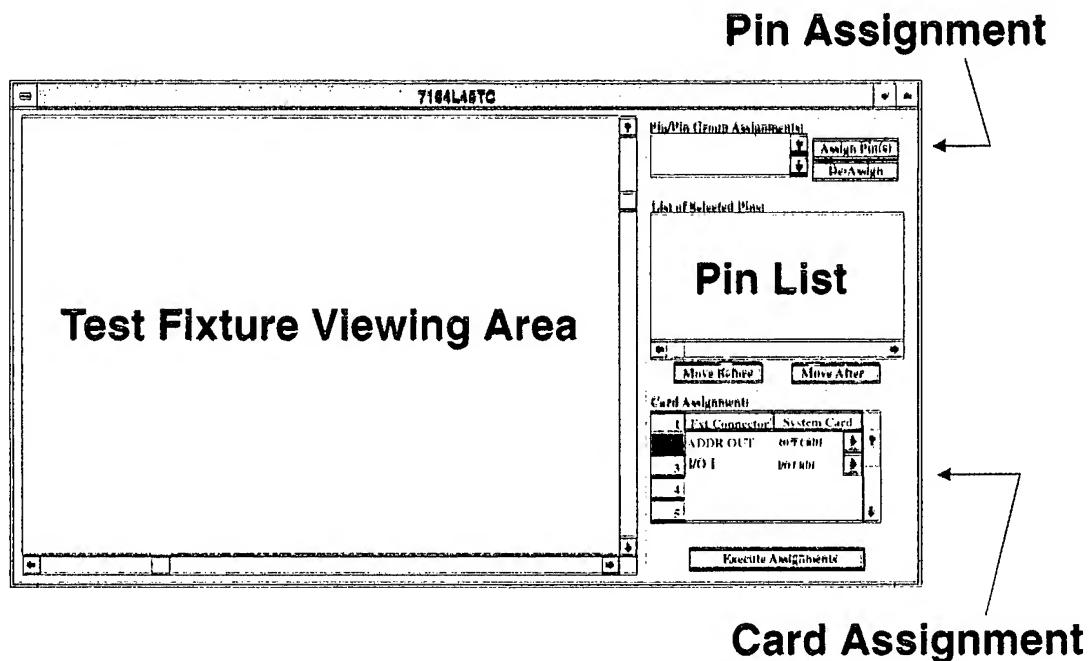

Figure 5-4. Test fixture and pin assignment.

24

5.2.1 Test Fixture Viewing Area.

The selected test fixture is displayed in the test fixture viewing in the dialog box as shown above. Initially, the test fixture is drawn at 100 percent resolution. Zooming in or out to different resolutions is accomplished by selecting **File|View|**. Available resolutions are: Fit In Window, 100, 125, 150 and 200 percent.

Creating a part pin assignment is accomplished by first assigning instrument cards to the test fixture connectors, then selecting pins and groups of pins and assigning pin names.

5.2.2 Test Fixture Connectors and Card Assignments.

To perform a test, loaded instrument drivers control instruments which are electrically connected to test cards. The user must specify the physical inter-connect between instruments and cards by assigning a test fixture connector to an instrument card.

A connector is a group of pins on the test fixture of all the same pin type. Connectors are determined by the designer of the test fixture. (See Creating A Test Fixture Section 1.3.1.1 and Test Fixture ASCII File Format Section 1.3.1.3). Typically, a graphical test fixture connector is the same as a physical connector on the test card, but it does not have to be. A physical test card connector may be subdivided into several graphical connectors, or a group of physical connectors may be implemented as one graphical connector, as long as the above criteria is met.

Instrument cards are defined by the instrument driver.

When making a new pin assignment, initially the test fixture pins are grayed and the pin hot spots are turned off. To make pins selectable, connectors must first be assigned to instrument cards in the **Card Assignment** group of the dialog box shown above. The center column of the Card Assignment group lists the connectors in this test fixture. The right column contains a drop down list which displays the instrument cards that are available to assign to the selected test fixture. Available cards are dependent upon the instrument drivers loaded in this session, and of those, which cards are compatible with the connector, and have not been previously selected. Compatibility is achieved when the instrument card has the same number of pins as the test fixture connector, and all the pins are of the same type.

Also included in the drop down list, is the **Locally Enable** option. This option enables a connectors pins without assigning an instrument inter-connect. This allows for the creation of generic pin assignments whose instrument connection are specified in a later application.

5.2.3 Making Pin Assignments.

Pin assignments involves naming pins and groups of pins on the test fixture to correspond to the part to be tested. These pin names are used by instrument drivers to properly stimulate the DUT. To make pin assignments, move the mouse cursor over a desired pin then press the left mouse button. The pin and its associated names become highlighted. To de-select the pin, click

25

elsewhere in the test fixture drawing area. Multiple pins are selected by depressing the **Shift Key** while clicking on the desired set of pins.

Once the pin or set of pins have been highlighted, assign a name by pressing the **Assign Pin(s)** button. A dialog box will popup requesting the name for this pin group. Enter a name, Press OK. The pin label names will be replaced by the new pin name. This name will also be placed in the **Pin/Pin Group Assignments** list in the dialog box as shown above.

If more than one pin was selected, the pin names will include an index number starting with zero. For example, if 4 pins were selected and given the name ADDR, the resulting pin labels would be ADDR0, ADDR1, ADDR2 and ADDR3. The index sequence of pins in a group will be the same order in which they were selected. In many testing applications this sequence is critical. To change the pin sequence for an existing group, see **Modifying Pin Groups** Section 5.2.5.

5.2.4   Pin List.

As you select pins, the details information is shown in the **List of Selected Pins** group of the dialog box shown above. The details are:

> User Assigned Name For the Pin (if one exists yet)
> Test Fixture Name For the Pin
> Test Fixture Connector this Pin Belongs to
> Instrument Card This Connector is Assigned to

Also, a the pin list can be displayed for an existing pin/pin group by double clicking the pin/pin group name in the Pin/Pin Group Assignments list.

5.2.5   Modifying Pin Groups.

The sequence of pins for an existing pin group can be modified from the **List of Selected Pins** group of the dialog box. This is accomplished by highlighting the desired pin to move in the sequence, then selecting either the **Move Before** or **Move After** buttons, and then double clicking on the pin list the name of the pin which the subsequent pin is to be moved before/after.

5.2.6   Printing Pin Lists and Test Fixtures.

You may make a hard copy record of the pin list along with the test fixture illustration by selecting **File|Print**. A dialog box will appear allowing you to selecting printing organized by pin group or by test fixture connector. You may additionally select whether or not to print the test fixture graphics. You may preview the pin list hard copy by selecting **File|Print Preview**.

5.3   MAKING TIMING DIAGRAMS.

A **Timing Diagram** is used to specify the timing for stimulation of the DUT. Each ATV pin has independent timing, and can store up to 7 timing sets which are selectable on-the-fly. Timing is specified by graphically creating timing diagrams similar to the ones found in part specification sheets.

Timing Diagrams are created in a dialog box which is accessed by selecting
Instruments|ATV|Timing Diagram|New or by selecting **Instruments|ATV|Timing Diagram|Open**. The dialog box illustrated below in Figure 5-5 will appear.
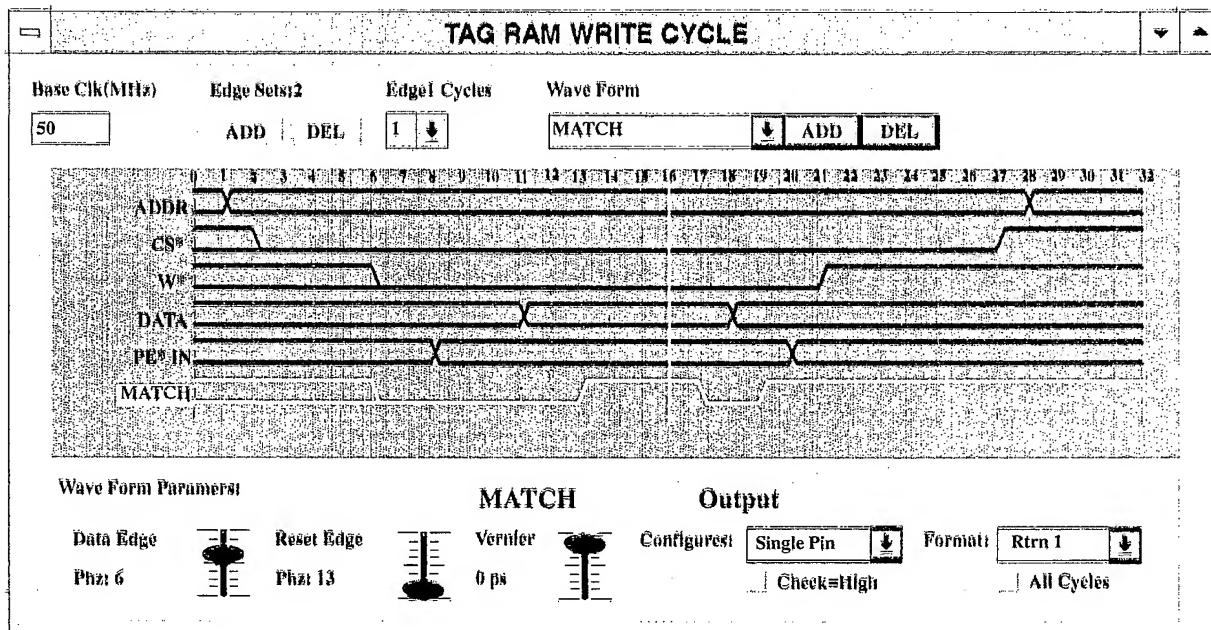

Figure 5-5. Timing diagram dialog box.

Testing involves outputting data or states to the part and reading data from the part in a specific timing. Output and compare data is specified in a test algorithms . Data can be generated algorithmically or can come from test vector files. (See Creating an Algorithm Section 5.4). Timing is specified in timing diagrams. A timing diagram is composed of **waveforms**. Waveform are given names which are matched to pin names from the part/package description to configure ATV pins. (See Test Fixtures Section 5.1).

A new waveform and its name are created by using the mouse to depress the **ADD** button shown in the above dialog box. A popup dialog box appears prompting for the name of the waveform. The popup also contains a list box where you specify the **Pin Type** as **Output** or **Cmpr.** Select an output pin type if this waveform is to assert data, or compare if it specifies the timing for which data will be read from the part. The new waveform is inserted directly below the waveform presently highlighted. A waveform can be highlighted by placing the mouse cursor

over the waveform illustration and pressing the left mouse button, or by selecting the waveform name from the list control in the WaveForm box. A waveform can be deleted by using the mouse to depress the **DEL** button.

Waveforms configure either **Single Pins** or **Pin Groups**. This is specified in the **Configures** list control of the **WaveForm Parameters** box shown above. When configuring single pins, the output data can optionally be specified in the timing diagram instead of an algorithm. Single pin output state is specified in the check box under the Configures control. (**Check=High**).

A waveform typically has a leading edge, called the **Data Edge**, and a trailing edge called the **Reset Edge**. A data and reset edge set define the timing in which data is output or compared. A timing diagram may contain as many as seven **Edge Sets**.

> (Note: A test may be composed of multiple timing diagrams, and can incorporate timing diagrams until seven edge sets are declared. For example, if a test uses a timing diagram that contains 5 edge sets, two edge sets remain for other timing diagram declarations. See Creating an Algorithm Section 5.5).

The number of edge sets is increased or decreased by using the mouse to depress the ADD and DEL buttons in the Edge Sets portion of the above dialog box. Edge set boundaries are illustrated with bold vertical white lines in the timing diagram.

Timing formats determine the state of the pin(s) at the completion of the data cycle. Timing formats include Return to One (**Rtrn 1**), Return to Zero (**Rtrn 0**), Return to Complement (**Rtrn C**), **NRZ**, or **No Change**. Formats are specified in the **Format** list control of the **WaveForm Parameters** box shown above. Formats are defined as follows:

Rtrn 1 -	Output goes to data value on data edge, goes to output high on reset edge
Rtrn 0 -	Output goes to data value on data edge, goes to output low on reset edge
Rtrn C -	Output goes to data value on data edge, goes to compliment of data value on reset edge
NRZ -	Output goes to data value on data edge and stays there, no reset edge
No Change -	No edges, values remain at ending state of previously applied timing diagram

The minimum **cycle** time for a timing diagram is set by the test frequency and can be as short as 20 ns. Each edge set can have a duration of 1 to 7 cycles. The maximum number of cycles definable in a timing diagram is 7 edge sets times 7 cycles each, for a total of 49 cycles.

Rough edge placement is 1/16 cycle boundaries. This gives down to 1.25 ns independent edge placement. Edges can be placed using either the slider controls or by dragging an edge using the mouse.

Fine edge placement is 100 ps resolution and is set with the **Vernier** slider control. Vernier moves both the data and reset edge together.

Blanking Pulse - ATV disallows an edge to be placed in the first phase of the first cycle of a timing diagram due to a hardware imposed blanking pulse.

Slider controls also respond to the up and down arrows for fine control. To use the up and down arrows, first left mouse click on the slider which you wish the arrows to control.

Vernier value is reported to the left of the slider control in actual time with units of picoseconds.

Edge timing is reported to the left of the slider control. By default, edge placement is reported in terms of cycle phase. You may optionally view edge placement in terms of **pseudo-time** by selecting **View|Change Units.** This causes edge placement to be reported in units of nanoseconds. Actual test frequency is specified at run time. Values presented next to the *Edge* slider controls are *pseudo-times* and given simply as an aid to timing diagram creation. The pseudo-times are based on the cycle frequency currently specified in the **Base Clk** control illustrated in the above. Edge placement time is the sum of edge value plus the vernier value.

Zoom - You many magnify a portion of the timing diagram for fine edge placement or reading by selecting **View|ZOOM IN.** This caused the curser to change to a magnifying glass shape when the mouse is over the timing diagram drawing area. Depress the left mouse button and drag left or right to create a zoom window. After the releasing the left mouse button, the timing diagram will redraw at the selected resolution. To zoom out, select **View|ZOOM OUT.**

All Cycle - If a timing diagram contains an edge set that has more than one cycle, a waveform may be optionally repeated for every cycle as long as the data and reset edge occur within a cycle boundary. This can be helpful in certain applications such as constructing clocks. This is specified in the All Cycles check box control as illustrated above.

5.4   ALGORITHM PROJECT.

An algorithm project combines a test algorithm with a test fixture and timing diagrams to create a test. It is also where algorithm source code is created and compiled and where download files are built for ATV. Algorithm Projects are accessed by selecting **Instruments|ATV|Timing Diagram|New** or by selecting **Instruments|ATV|Timing Diagram|Open.** The dialog box illustrated below in Figure 1-15 will appear.

An algorithm project builds a group of three download files all with the same name but different extensions (***.AXT, *.AXF, *.AXG**) defined as follows:

    *.AXT - Opcode file containing algorithmic binary code for the ATV processor.
    *.AXF - Timing code file for output pins.
    *.AXG - Pin grouping and type information file.

The file name is indicated in the **Executable Names** text box at the top of the algorithm dialog box. These files are used to configure the ATV for test at run time. (See Test Execution 5.5).

The algorithm project dialog box is subdivided into Source Code, Text Fixture and Timing Diagram groups. The following presents how to use controls in each of these areas. Detailed information on creating algorithms and building download files is given in Creating an Algorithm.
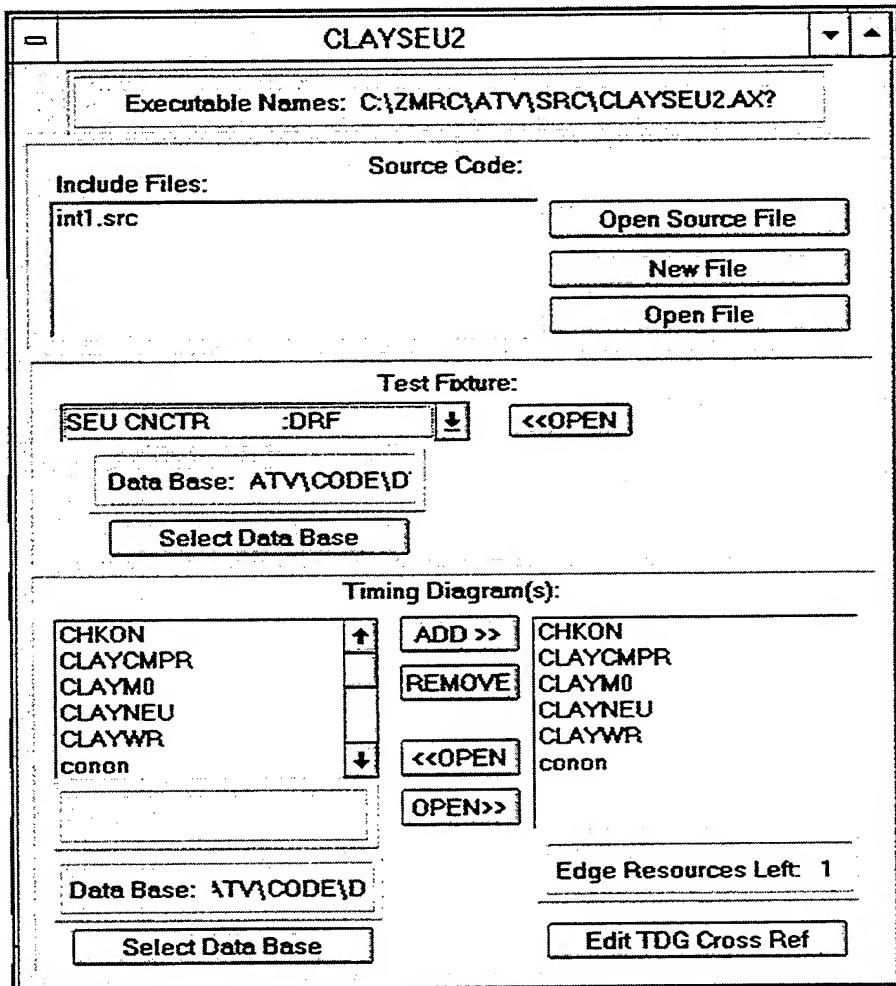
Figure 5-6. Algorithm project dialog box.

5.4.1 Source Code.

Source code for the algorithm is from a file with the executable name and a *.SRC extension. It can be accessed for editing automatically by using the mouse to depress the **Open Source File** button shown in the above dialog box. Any include files declared in the source file are listed in the **Include File** list box control. Include files can be automatically opened by double clicking on the include file name. The algorithm project also allows for the creating and opening of any ASCII text file by using the **New File** and **Open File** buttons.

5.4.2 Test Fixture.

The test fixture to be used with this algorithm is selected in the **Test Fixture** group. The drop down list presents all the test fixtures available from the current data base. Use the **Select Data Base** button to view test fixtures from an alternate data base. Select a test fixture to be used for this algorithm by highlighting it's name in the drop down list. View the test fixture by using the mouse to depress the **<<OPEN** button.

5.4.3 Timing Diagram(s).

The timing diagrams to be used along with this algorithm are selected in the **Timing Diagram(s)** group. The left panel list control presents the timing diagrams available from the current data base. Use the **Select Data Base** button to view test fixtures from an alternate data base. Select timing diagrams to attach to this project by using the **ADD>>** button. An algorithm test can have up to 7 edge set resources. (See Making Timing Diagrams Section 5.3). As timing diagrams are added to this project, the remaining available resources are reported in the **Edge Resources Left** box.

The algorithm source code declares place holders for some number of timing diagrams. Timing diagrams specified in this project are linked into these place holders when the ATV download files are built. (See Creating an Algorithm Section 5.5). By default the name used in the algorithm is expected to be the name shown in the timing diagram list. To use timing diagrams with names that do no match the algorithm declarations, make a timing diagram cross reference by using the mouse to depress the **Edit TDG Cross Ref** button.

5.5 CREATING AN ALGORITHM.

Creating an algorithm involves writing source code, compiling it with timing diagrams selected in the algorithm project, then building ATV download files. This section describes the details of how to create a new algorithm.

Create a new Algorithm Project by selecting **Instruments|ATV|Timing Diagram|New.** The algorithm project dialog box appears. (See Algorithm Project Section 5.3). Use the mouse to depress the **Open Source File** button. A dialog box appears prompting for the **Source Code Base Directory** for this project and the name for the **Executable File Name** set. The algorithm project creates four files with the same name and different extensions. (See Algorithm Project

31

Section 5.4). The executable files will be placed in the base directory. To have the files placed in the current directory leave the base directory blank. Enter the path, and an executable name with no extension and press Enter. The text editor appears with template source code as shown below in Figure 5-7:

All ATV algorithms must include the 'interupt.src' file which sets up the interrupt routines for the processor. A default interupt.src is shipped along with ATV. Move a copy of interupt.src to the base directory for this project.

Next you will write and debug algorithm source code. Language elements and examples are given in the Programmers Guide. This example will use the code presented below in Figure 5-8 to complete the discussion of algorithm creation.
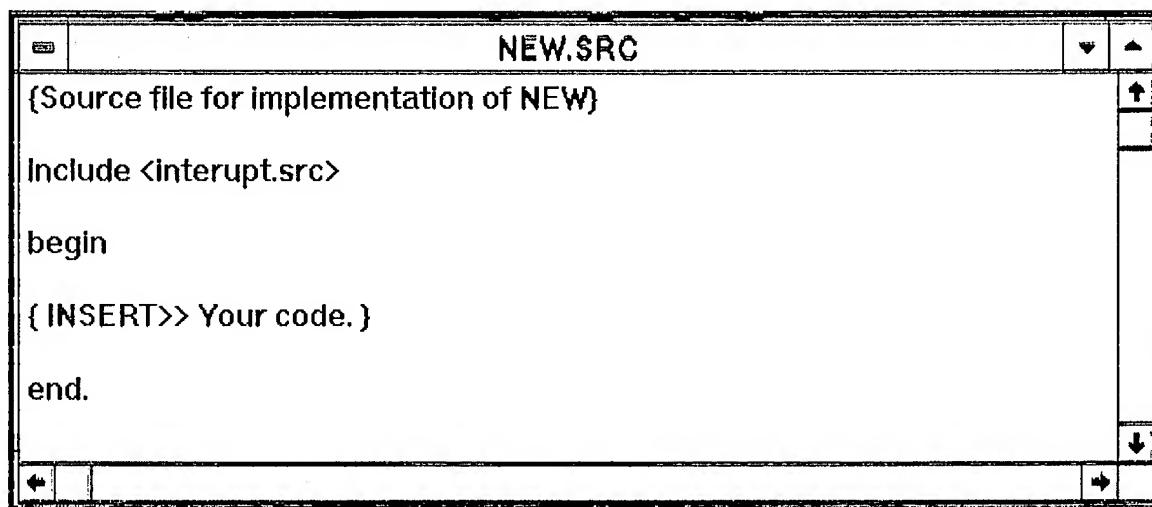
```
NEW.SRC

{Source file for implementation of NEW}

Include <interupt.src>

begin

{ INSERT>> Your code. }

end.
```

Figure 5-7. Text editor with ATV algorithm template code.

```
oneszros.SRC

{Source file for implementation of oneszros}

include <int1.src>
label LP1, LP2, LP3, LP4,LPST;
varINT vhid1, vlod1, vhid2, vlod2, vvhi, vvlo, vvdd, vidq, I;

varTDG WR, RD;

varWVF DATA;
varADR ADR;

{PROCEDURE TO SET TVP CARD OUTPUT LEVELS, VDD, AND IDDQ REFERENCE}
procedure zdaci(varINT vhid1,vlod1,vhid2,vlod2,vvhi,vvlo,vvdd,vidq);
        begin

        inline(
                MOV TOP vhid1;
```

Figure 5-8. Sample algorithm code.

Select timing diagrams appropriate for your algorithm as described in Algorithm Project. Timing diagram selection is dependent upon declarations in the source code. The source code declares **varTDG** variables which are place holders for timing diagrams. In the above example two such variables are shown; WR and RD, indicating that two diagram are needed for this project. Also the source code declares **varWVF** and/or **varADR** waveform variables for the selected timing diagrams. In this example the DATA and ADR waveforms of the WR and RD timing diagrams will be used. Therefore, two timing diagrams must be selected for this example project with the names WR and RD and they must include waveforms named DATA and ADR. (Note: The timing diagrams can have different names than those declared with varTDG if a cross reference is used. See Algorithm Project Section 5.4).

As the code is being written, compile and debug it by selecting **Algorithm|Compile.** A dialog box appears reporting the status of compilation. If errors exist an **Error Message** box appears. Double clicking on an error in the Error Message Box causes the line containing the error in the source code to be highlighted.

It is possible to view the assembly code that is resulting from the compilation by selecting **View|SetView.** A dialog box appears allowing for the selection of viewing assembly after PAS1, PAS2 or Link, and for viewing Mixed Assembly and source code or assembly only. When the assembly code is veiwed, it is also placed in a *.**ASM** file with the executable name for subsequent viewing.

After the algorithm successfully compiles, the next step is to build the executable file set for down load to the ATV processor. First select the test fixture that supplies pin names as described in Algorithm Project . Next select **Algorithm|Build DownLoad.**

Checking the **PASS/FAIL** box will cause ATV to report a pass or fail result in the data log for each test. Checking the **Log Group Errors** box will cause ATV to report detailed pin failure information. The **Max Errors to Log Per Test** edit control specifies a value after which error reporting is truncated.

Test Counter Initial Value controls the initial test number reported in the data log. **Part Serial Number** is composed of an alpha and a numeric string. By checking the **Auto Increment** box the numeric portion will be automatically incremented on subsequent tests.

5.6.2   Loading and Running the Algorithm Test.

Prior to running an algorithm test the executable files must first be downloaded to the ATV processor. To load the test select **Algorithm|Load.** Next run the test by selecting **Algorithm|Run.** The **Run Algorithm** dialog box shown below in Figure 5-10 appears.
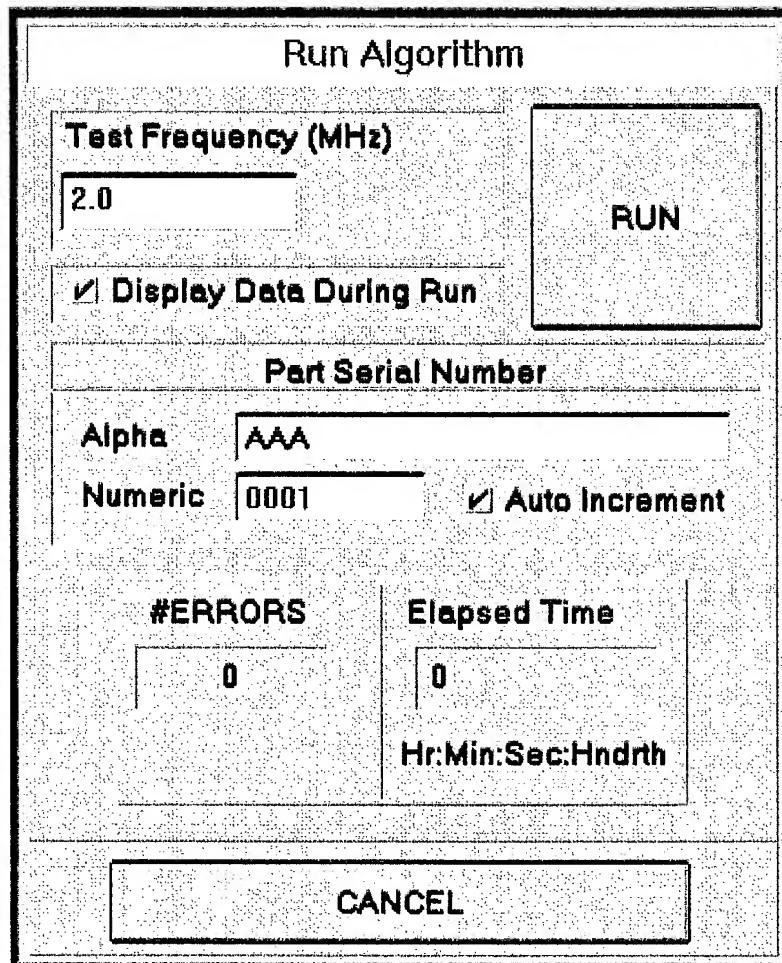
Figure 5-10. Run algorithm dialog box.

35

SECTION 6
PROGRAMMERS GUIDE

LANGUAGE STRUCTURE.

ATV achieves its high performance and flexibility in large part because of the close coupling between its hardware and software. The Test Vector Processor (TVP) is a custom designed RISC processor with special features for maximizing the speed of test execution. Because of its unique design, a special set of assembly language instructions was developed for programming the processor. A custom compiler processes the assembly instructions and produces the appropriate operational codes (op codes) to be downloaded into the TVP memory.

A custom high level language was also developed which is a subset of PASCAL. This language incorporates all major aspects of modern programming languages, including procedures with parameter passing, basic branching operators, conditionals and math expressions. The language also implements an include operator which allows source code from other modules (documents) to be linked at compile time. Therefore, algorithm and language libraries can be built by MRC, or the user, to extend the language and provide basic device test functions such as GALPAT, Marching 1's, etc. The language is described in Figures 6-1 through 6-4 as syntax flow charts.

Mixed level programs can be written through the use of an 'inline' operator within the high level language (i.e., the high level language allows inline assembly).

The compiler has comprehensive error checking for lexical (typographical), syntatical and semantic type errors. The compiler can be optionally set to display the intermediate code which is generated from the high level source code. This can be helpful to the user in optimizing algorithms. For example, viewing assembly level might be used to optimize code so test devices are exercised at the highest possible speeds by minimizing processing steps between outputs to the device.
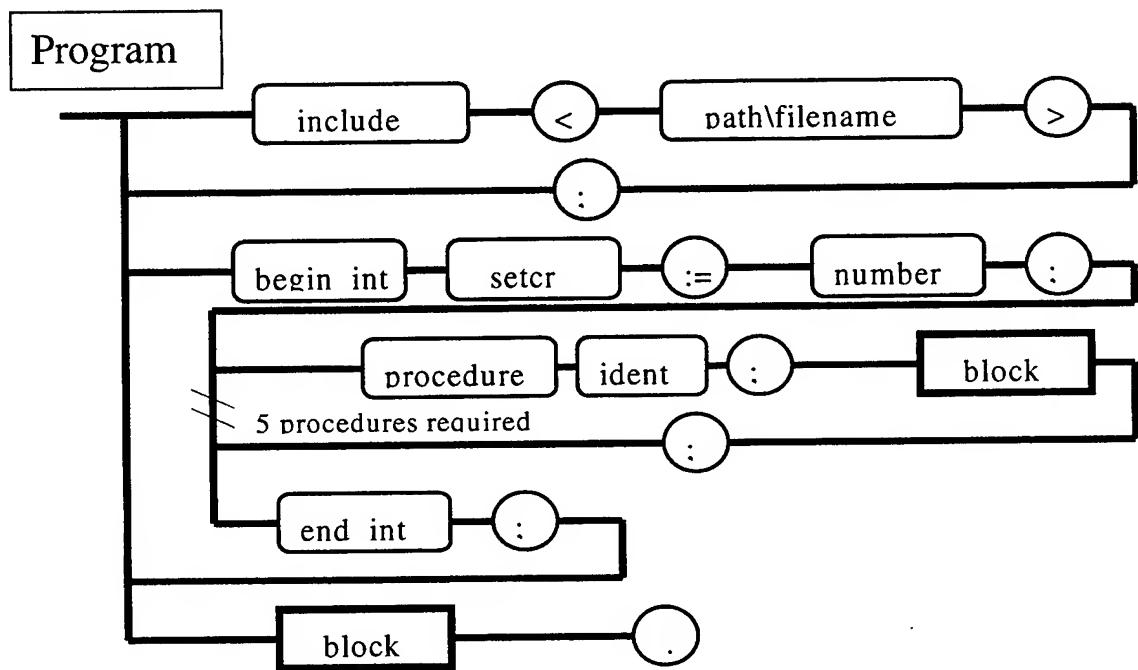
Figure 6-1. Syntax chart of program flow.
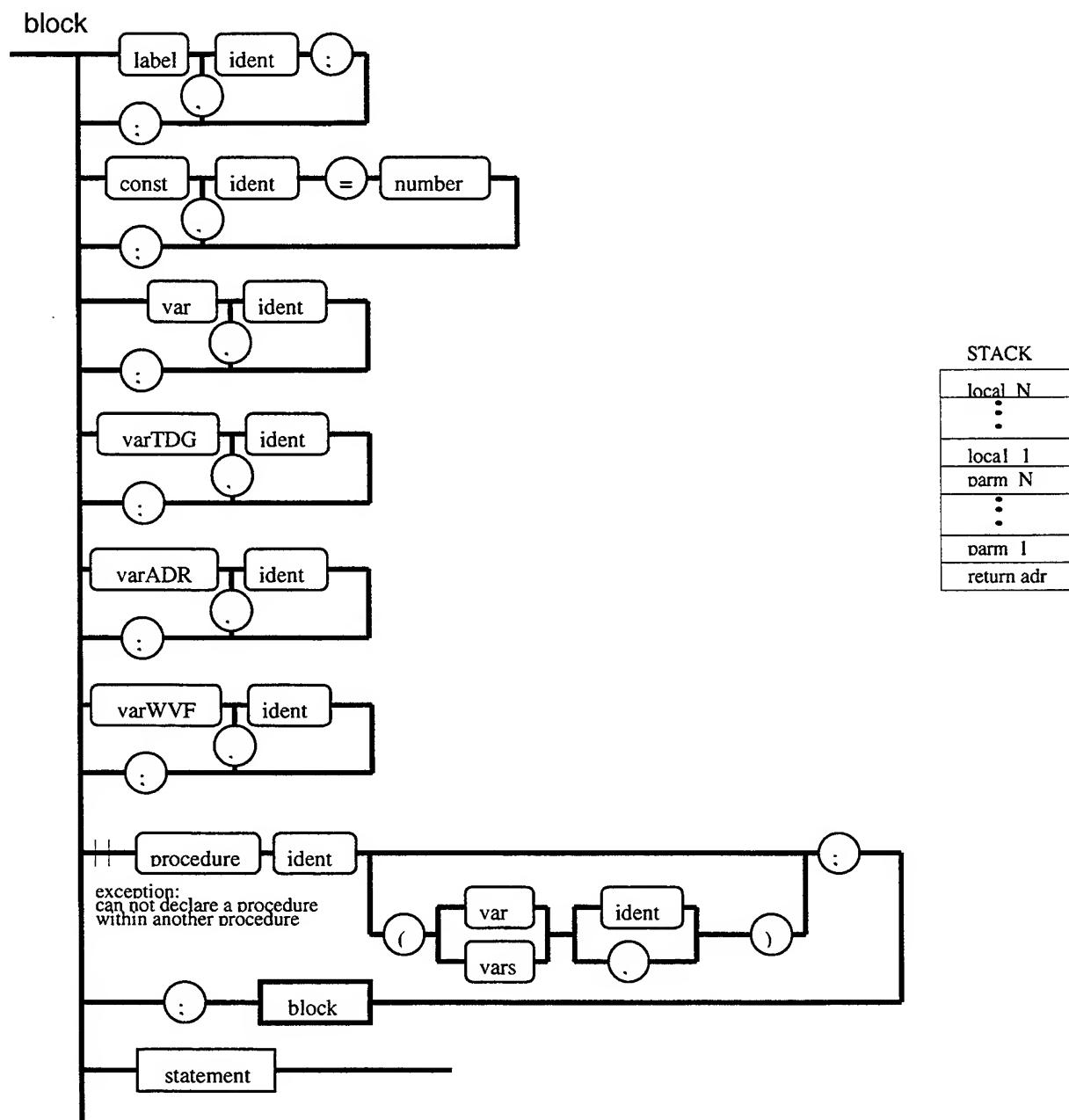
Figure 6-2. Syntax chart of a program block.

39

statement
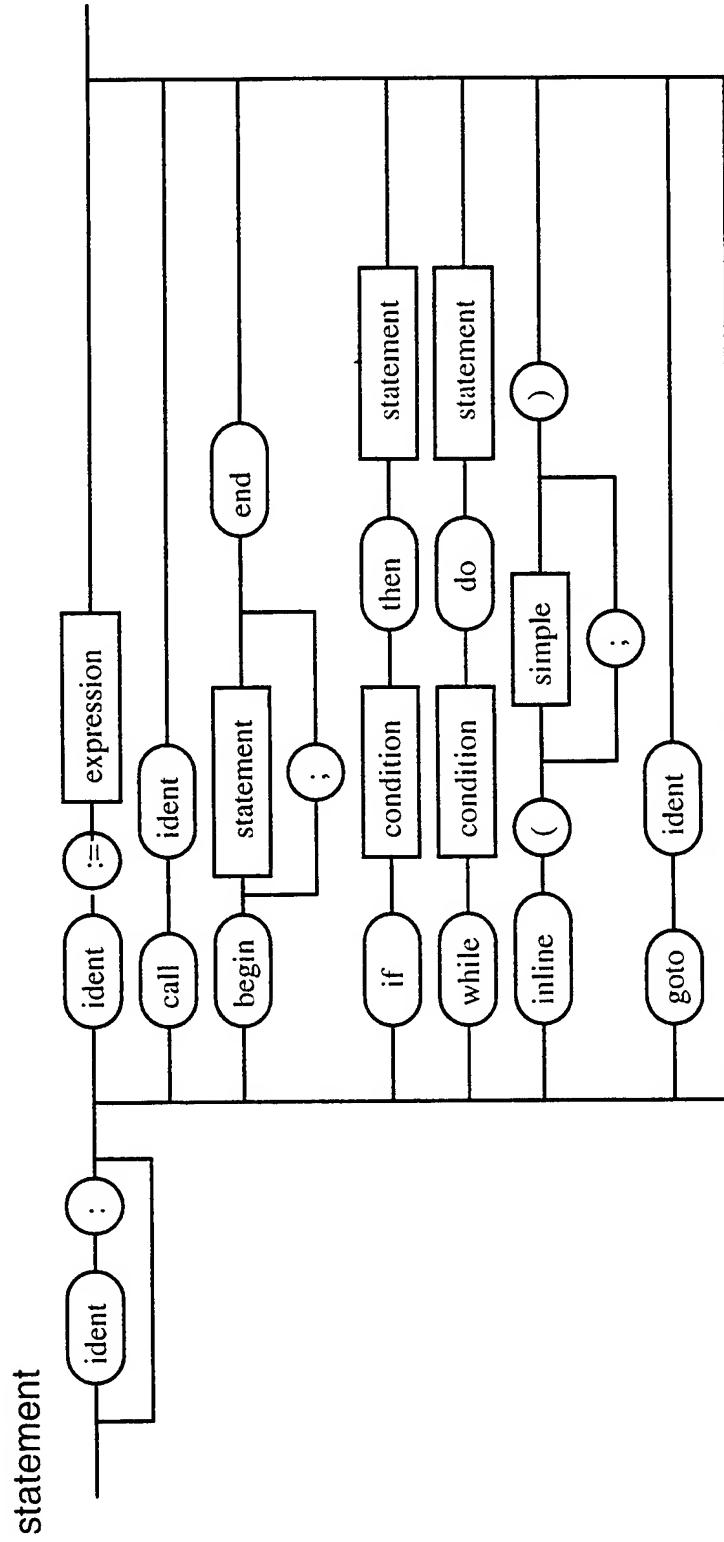
Figure 6-3. Syntax chart for high level instructions.

40

simple

Figure 6-4. Syntax chart for assembly level instructions.

41
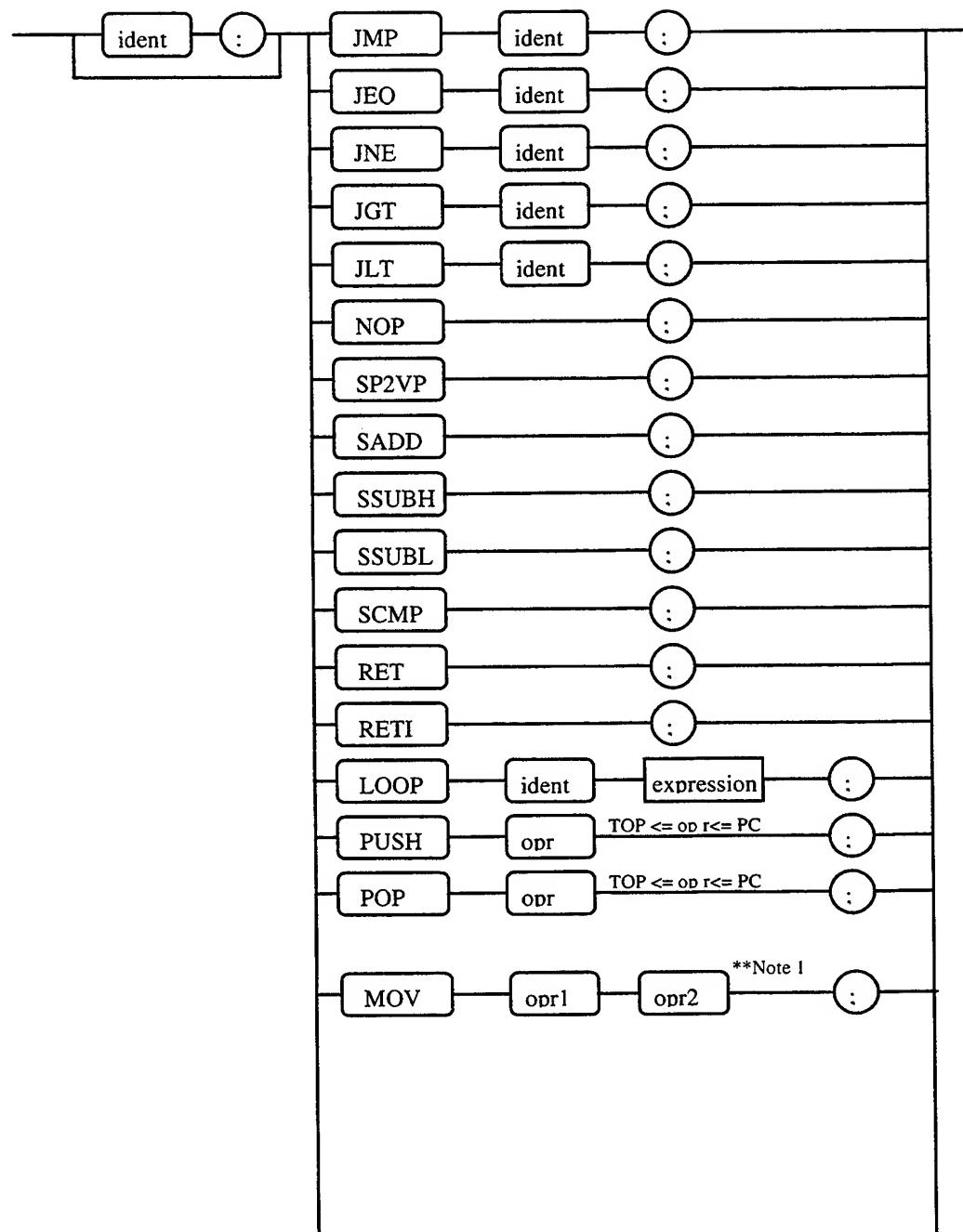
simple
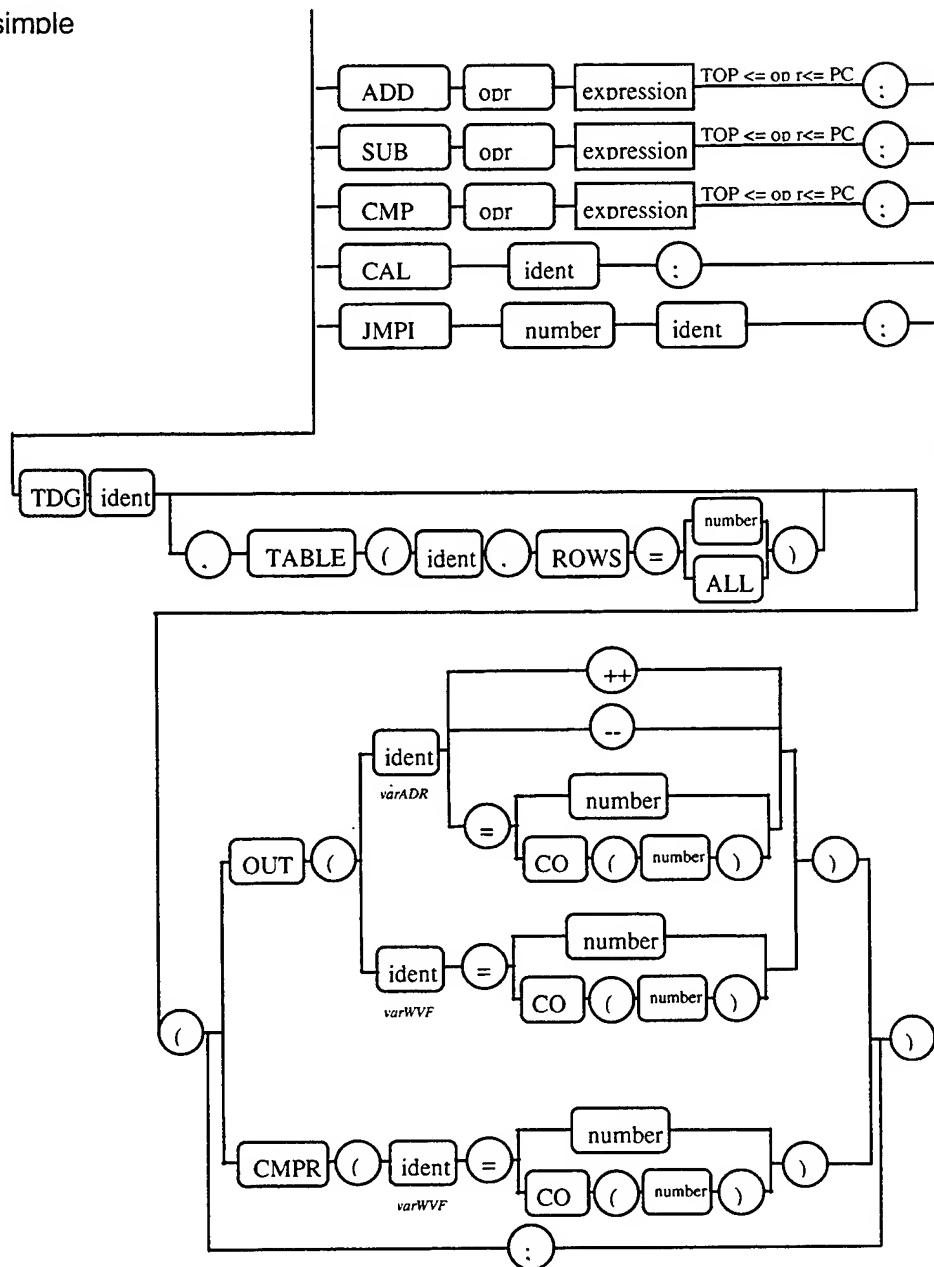
$$opr = \{[VP], TOP, BOT, EA, SA, CX, ERCNT, VP, AX, PC, SP, CR\}$$

**Note 1:   if opr 1 = [VP] then $TOP <= opr2 <= PC$

if $TOP <= opr1 <= AX$ then $opr2 = \{ [VP], number, expression \}$

Figure 6-4. Syntax chart for assembly level instructions. (continued)

SECTION 7
FIELD TEST RESULTS FOR NATIONAL SEMICONDUCTOR CLAY DEVICES

7.1   INTRODUCTION.

Single event effects (SEE) tests on the National Semiconductor Corporation Configurated Logic Array (CLAy) chip were performed at the Brookhaven National Laboratories Tandem Van de Graaf (TVG) on 11 June 1996 using the MRC developed Algorithmic Test Vector (ATV) Memory and ASIC Test System.

The Configurable Logic Array Field Programmable Gate Array (FPGA) is a CMOS-SRAM based high density device. The CLAy devices use an array of cells connected to a flexible bussing network. The array is surrounded by programmable I/O.

The CLAy is fabricated using the National Semiconductor Corporation CS065 single-poly, multi metalization, CMOS process shown in Figure 7-1. The test devices were mounted in Pin Grid Array packages with removable lids for testing at the Brookhaven Tandem Van de Graaf Facility. This package allowed for the attachment of a heater and resistance thermal detector (RTD) to the bottom of the package. Thus elevated temperature tests could be performed.

The test devices have 64,512 bits of storage and were programmed with a modified checkerboard pattern. As many bits were placed in alternating states consistent with correct operation of the device. The algorithm used for programming the pattern into the CLAy device is given in Table 7-1.
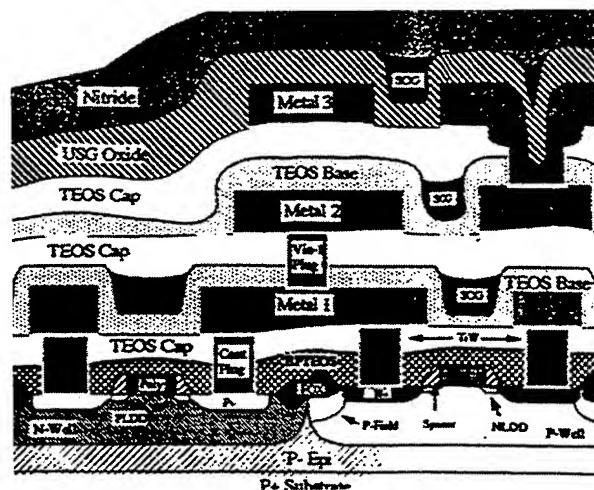
Figure 7-1. CS065 Cross Section.

43

TABLE 7-1. TEST ALGORITHM FOR CLAY FPGA.

Preamble and misc setup stuff (12 bytes)

0x00,	0xb2,	0x00,	0x00,	0x00,
0xFE,		0x7d,		0x3F

Next 6272 bytes (112 bytes per row times 56 rows)

|				
first row	0x11,	0x00	followed by 55 pairs of 0x84,	0xD1
second row	0x14,	0x80,	followed by 55 pairs of 0x84,	
third row	0x22,	0x00	followed by 55 pairs of 0x84,	
fourth row	0x14,	0x80,	followed by 55 pairs of 0x84,	
fifth row	0x22,	0x00	followed by 55 pairs of 0x84,	
sixth row	0x14,	0x80,	followed by 55 pairs of 0x84,	
seventh row	0x22,	0x00	followed by 55 pairs of 0x84,	
eighth row	0x14,	0x80,	followed by 55 pairs of 0x84,	
	***			
36th row	0x14,	0x80,	followed by 55 pairs of 0x84,	
	***			
50th row	0x14,	0x80,	followed by 55 pairs of 0x84,	
51st row	0x22,	0x00	followed by 55 pairs of 0x84,	
52nd row	0x14,	0x80,	followed by 55 pairs of 0x84,	
53rd row	0x22,	0x00	followed by 55 pairs of 0x84,	
54th row	0x14,	0x80,	followed by 55 pairs of 0x84,	
55th row	0x22,	0x00	followed by 55 pairs of 0x84,	
56th row	0x04	0x80,	followed by 26 pairs of 0x01,	0x81, 0x04, 0x08, 0x81

followed by 0x01,   0x04,   0x81,   0x81)

(112*6)

Next 672 bytes

first row	0x00,	0x00,	followed by 55 pairs of 0x004,	0x04,
second row	0x00,	0x00,	followed by 55 pairs of 0x004,	0x04,
third row	0x00,	0x00,	followed by 55 pairs of 0x004,	0x04,
fourth row	0x00,	0x00,	followed by 55 pairs of 0x004,	0x04,
sixth row	0x00,	0x00,	followed by 55 pairs of 0x004,	0x04,

0x01
0x01
0x08

Next 672 bytes: 660 0x00's followed by 6 pairs of 0x00,
Next 112 bytes: 0,   0,   followed by 55 pairs of 0x00,
Next 112 bytes: 0,   0,   followed by 55 pairs of 0x00,
Next 56 bytes 0x00
Next 56 bytes 0x00 followed by 27 pairs of 0x04, 0x09   followed by 0x00
Next 112 bytes 0x00
Next 2 bytes of post amble 0x4d

44

7.2   TEST CONFIGURATION.

The CLAy chips were mounted on a circuit board designed especially for the Brookhaven heavy ion vacuum chamber. The circuit board supported three (3) CLAy devices, thus allowing for the test of all three of the devices without breaking vacuum. The cables from the circuit board were connected to vacuum feed through connectors in the bottom plate of the vacuum chamber, and in turn were connected to the ATV system.

The devices were mounted in pin grid array sockets and included a heater and temperature sensor under the device chip. The lids of the devices were removed from the test. Latchup tests were performed at 80°C with a supply voltage of 5.25 volts. Some calibration tests were made at other temperatures and voltages, however, these were not included in the latchup data. Single Event Upset (SEU) tests were performed at nominal device temperature (~29°C) and a supply voltage of 4.75 volts.

7.3   TEST MEASUREMENTS.

The system was functionally tested before evacuating the chamber. After assuring proper operation, radiation testing began. The MRC-developed Algorithmic Test Vector Test System (ATV) memory and ASIC test system was used for these tests. The ATV is a portable tester designed to perform high speed logic and timing tests of memories and ASIC's in the radiation environments. It can perform tests at 50 MHz and can control timing to within 100 picosecond edge placement. It has independent pin timing, fully adjustable input and output drive levels, and "on-the-fly" selectable timing. Data patterns can be generated algorithmically or can be imported from a separate file.

High speed testing of remote DUTs can be performed without any buffer electronics being located at the DUT. A resistor is placed in series with the DUT outputs for cases where a DUT can not directly drive the load presented by a coaxial cable. The system software automatically adjusts data comparison levels to account for signal attenuation caused by any series resistor.

The ATV provided the stimulus for the parts tested and logs logged radiation induced upsets. It recorded and displayed the results. Before irradiation, the ATV wrote a known pattern into the DUT, and read data back to verify that the part was working properly. After exposure began, the ATV periodically tested the DUT to determine if an upset had occurred. If an upset occurred, the ATV recorded the time and location of the upset in an Excel format spread sheet and wrote new data to the DUT.

The DUTs were typically sampled four times a second for typical runs of 300 seconds. The sample time is on the order of milliseconds, and is significant with respect to the total experimental time. For upset tests, the particle flux was experimentally determined to keep the error rate to approximately one per second in order to assure that all upsets were counted. The flux for the tests was kept at approximately 5×10^3 particles per second or less.

45

7.4 DATA.

7.4.1 Upset.

Three devices were tested for Single Event Upset. These devices have been identified as CLAy1, CLAy2, and CLAy3. A full characterization of the devices was made throughout the spectrum of available ions.

Table 7-1 displays the composite upset data taken for the three CLAy devices. A characterization was made for each of the devices from the threshold LET and continued to higher LETs where the saturated cross section was observed.

Individual plots of the upset data are given in Figure 7-2. These are raw data plots and no specific curve fitting was attempted for these data. Figure 7-3 shows the composite data for the three CLAy devices tested for upset. A Wiebul curve fit has been made to the composite data and the results of the curve fit is also given in Table 7-2.

7.4.2 Latchup.

Four devices were tested for single event latchup. The tests were made up to a LET value of 90.5 MeV-cm^2/mg (gold at 45°), and a fluence of 1E5 for that particular test. The device temperature for the latchup tests was maintained at 80°C. Latchup was not observed during any of the tests performed on the devices.

The maximum total fluence for CLAy4 at a LET of 71.4 MeV-cm^2/mg was 1.13E6, which would imply that the latchup cross section must be less than 9E-7 cm^2/device.

The maximum fluence to which a device was exposed depended on the flux used, and the number of upsets that were experienced. If the flux was too great, the number of upsets during the latchup test became enormous. The risk of not measuring errors and an enormously high error rate became unrealistic for the elevated temperature and increased voltage tests. Therefore a low flux (~10E3 p/s) was required. A test to 1E6 required an exposure between 5 and 10 minutes. An exposure to 10E7 would have required between 50 and 100 minutes, which was impractical for the experimental time available.

7.4.3 Data Integrity.

The ATV System stored the logical address of each error. This provided an easy way to examine the data to insure the errors were occurring randomly. Figure 7-4 is a plot of the logical location of the error from Run 15.

Table 7-2. Single Event Upset Data for NSC CLAy Devices

CLAy1

UPSET TESTS								
Run No.	Ion	Energy at SI Surface MeV	LET MeV-cm²/mg	Incident Angle Degrees	Fluence Ions/cm²	Number Upsets	Cross Section Cm²	Accum. Dose
36	FL-9	143.6	3.3	0	5.02E+04	1	1.99E-05	5.43E+02
33	FL-19	143.6	3.33	0	1.02E+04	1	9.79E-05	5.36E+02
35	FL-19	141.6	4.38	40	5.03E+04	23	4.58E-04	5.40E+02
27	CL-35	187.3	12.1	0	1.01E+04	45	4.46E-03	5.33E+02
32	CL-35	180.1	16	40	1.13E+04	57	5.04E-03	5.35E+02
21	BR-79	213.5	38.5	0	2.12E+04	131	6.19E-03	5.31E+02
7	I-127	210.6	57.6	0	5.12E+04	84	1.64E-03	3.87E+02

CLAy2

UPSET TESTS								
Run No.	Ion	Energy at SI Surface MeV	LET MeV-cm²/mg	Incident Angle Degrees	Fluence Ions/cm²	Number Upsets	Cross Section Cm²	Accum. Dose
37	FL-19	143.6	3.33	0	5.02E+04	2	4.00E+02	1.19E+02
38	FL-19	141.6	4.38	40	5.02E+04	14	2.92E+02	1.23E+02
26	CL-35	187.3	12.1	0	1.10E+04	44	2.51E+02	1.14E+02
30	CL-35	180.1	16	40	1.01E+04	53	1.93E+02	1.17E+02
25	BR-79	213.5	38.5	0	9.93E+03	134	6.25E+01	1.12E+02
17	I-127	210.6	57.6	0	1.02E+04	150	1.64E+02	9.03E+00
18	I-127	210.6	57.6	0	5.54E+03	83	2.69E+02	9.41E+01

CLAy3

UPSET TESTS								
Run No.	Ion	Energy at SI Surface MeV	LET MeV-cm²/mg	Incident Angle Degrees	Fluence Ions/cm²	Number Upsets	Cross Section Cm²	Accum. Dose
39	F-19	143.6	3.33	0	5.01E+04	2	399E-05	1.38E+02
40	F-19	141.6	4.38	40	5.01E+04	10	2.00E-04	1.42E+02
28	CL-35	187.3	12.1	0	1.01E+04	40	3.62E-03	1.33E+02
29	Cl-35	180.1	16	40	1.01E+04	59	5.87E-03	1.36E+02
24	BR-79	213.5	38.5	0	1.01E+04	164	1.63E-02	1.31E+04
15	I-127	218.6	57.6	0	1.10E+04	106	1.69E-02	1.16E+02

| WIEBUL CURVE FIT PARAMETERS USED				
	CS 1.5E-2	LO 3	W 18	S 2
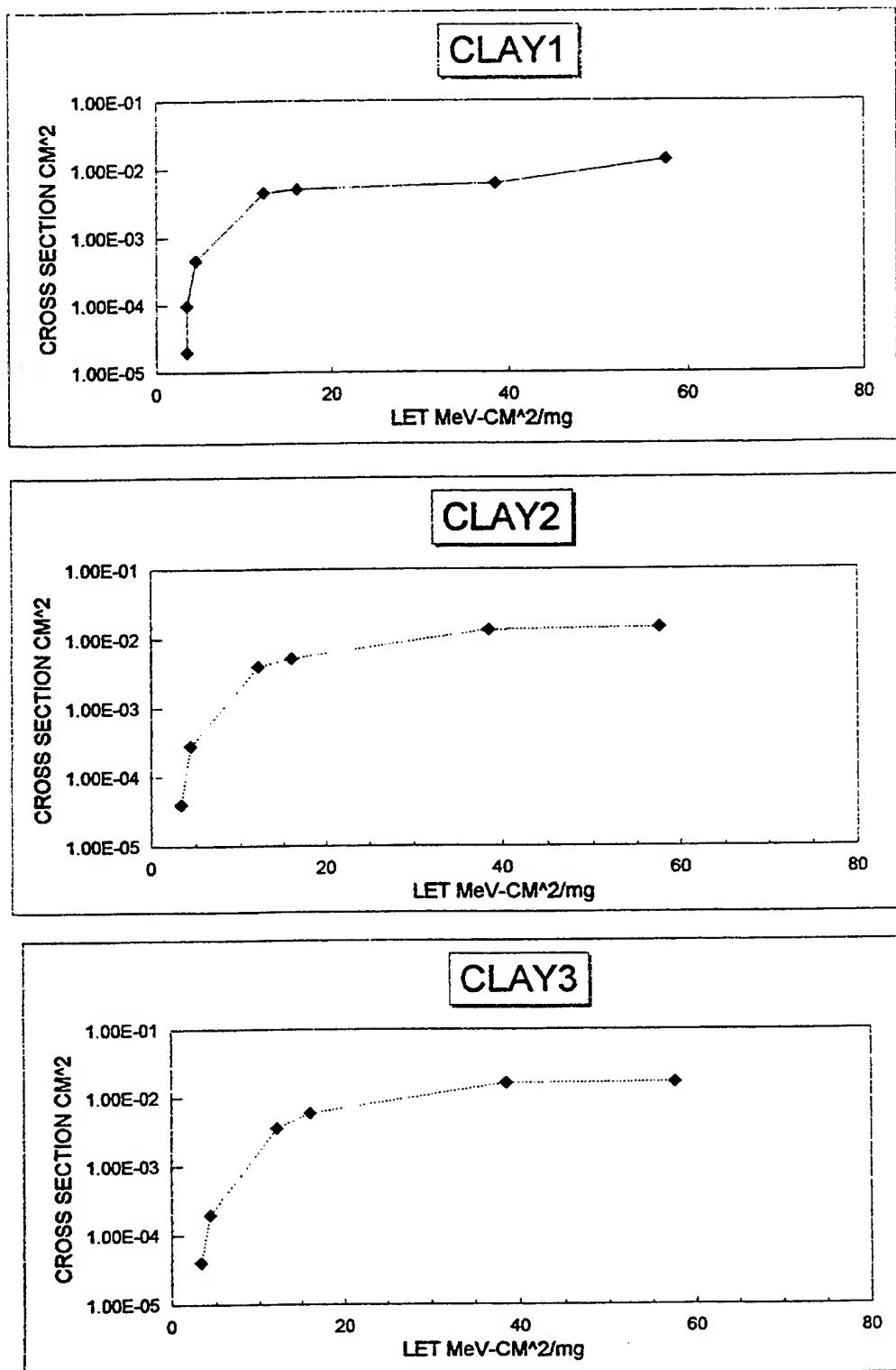
Figure 7-2. Plots of SEE Upset Data for the CLAy Chip.
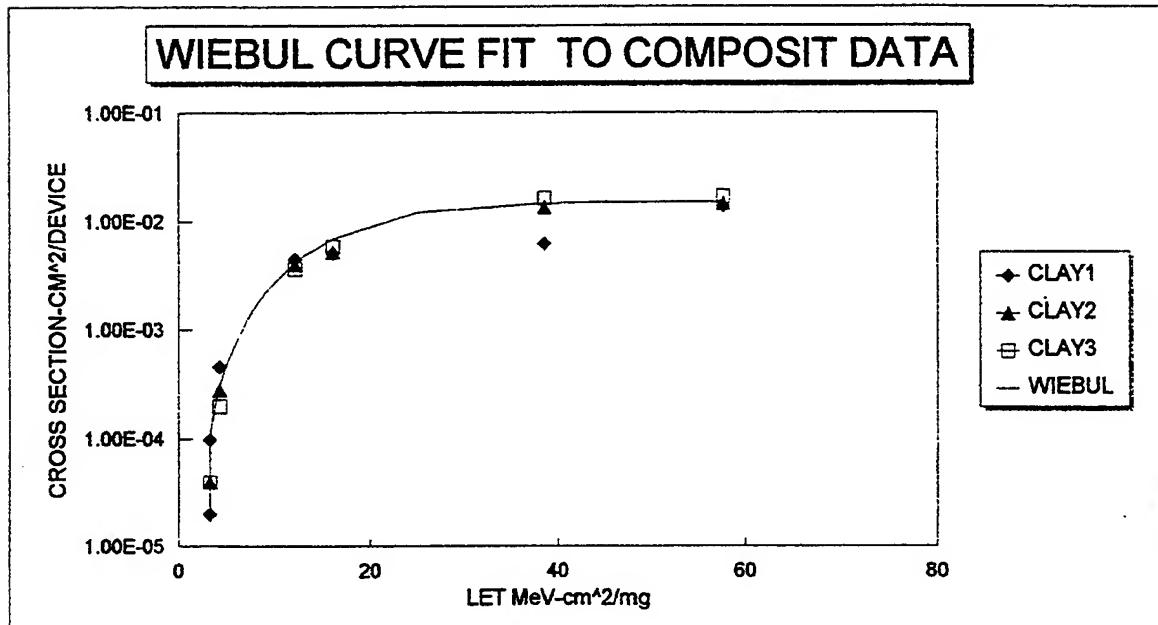
Figure 7-3. Composite data for the three CLAy devices tested for upset.
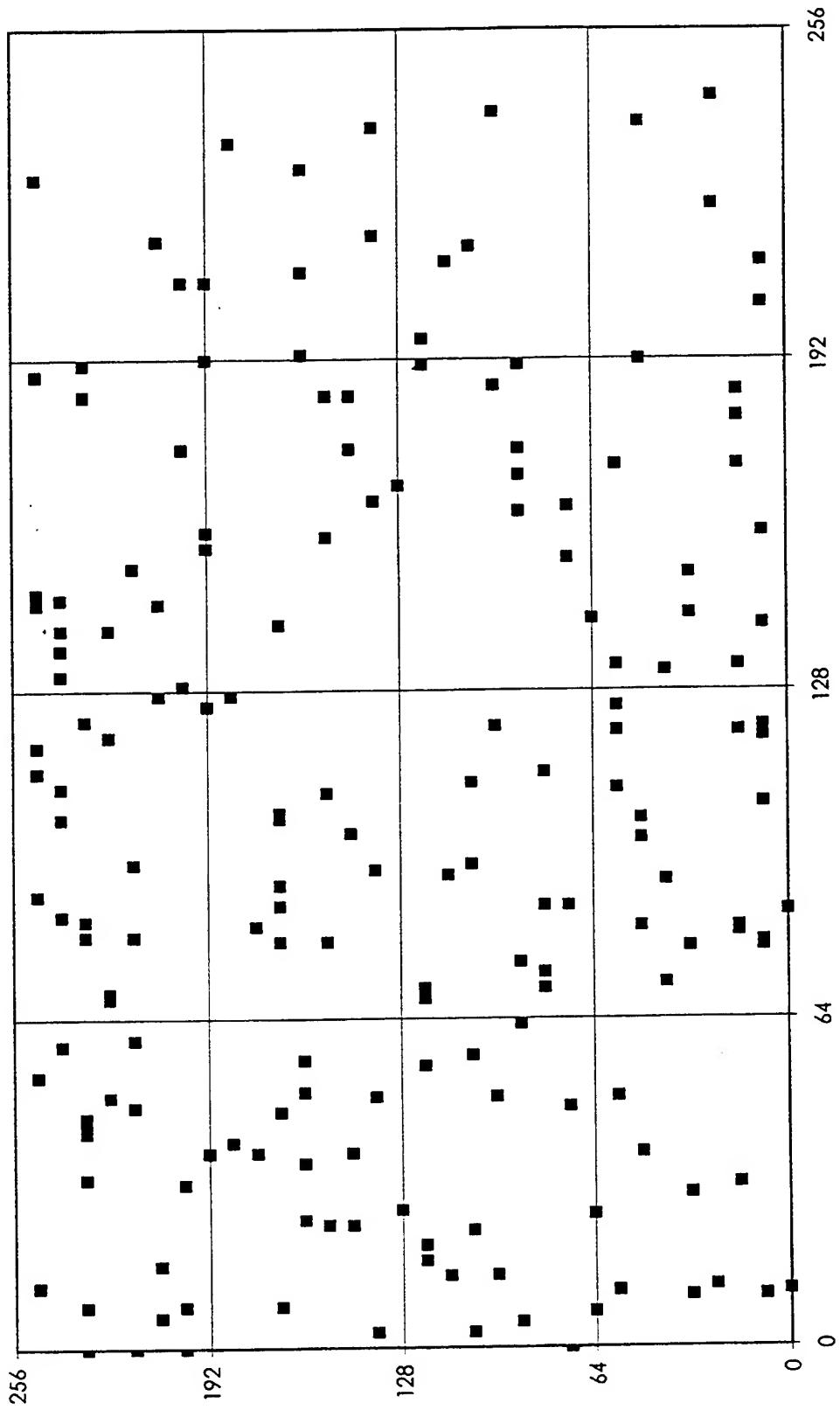
Figure 7-4.   Logical memory map of CLAy SEU upsets for Run 15, 186 errors in 79 seconds ... 2.3.5 errors/seconds.

7.5   CONCLUSION.

The results of these tests provide the following conclusions.

1.   The upset data indicates a reasonable SEU immunity for a commercial device which is not specifically hardened for these effects. The SEU results indicate that the device may be acceptable for many applications in low earth orbit. The results also indicate that the application in the more harsh environment of the geosynchronous orbit may also be acceptable with some provision for 10 to 20 upsets per device per year.

2.   The latchup data indicates that the device is relatively insensitive to latchup. The devices showed no indication of latchup under any test conditions. The specific latchup tests performed at 80°C and a supply voltage of 5.25 volts showed no latchup up to a fluence of 1.12E06 at a LET of 71.4 MEV-cm^2/mg, and no latchup up to a fluence of 5E04 at a LET of 90.5 MeV-cm^2/mg.

3.   The ATV System worked well and provided a simple way to test a very complex part.

DISTRIBUTION LIST

DEPARTMENT OF DEFENSE

DEFENSE ADVANCED RESEARCH PROJECTS
AGENCY
3701 NORTH FAIRFAX DRIVE
ARLINGTON, VA 22203-1714
   ATTN: ASST DIR (ELECTRONIC SCIENCES DIV)

DEFENSE TECHNICAL INFORMATION CENTER
8725 JOHN J KINGMAN RD., SUITE 0944
FORT BELVOIR, VA 22060-6218
   ATTN: DTIC/OCP

DEFENSE THREAT REDUCTION AGENCY
6801 TELEGRAPH ROAD
ALEXANDRIA, VA 22310-3398
   ATTN: CPWCT
   ATTN: NSSA
   ATTN: NSSA, W SUMMA
   ATTN: NSSE, L PALKUTI
   ATTN: NSSE, L COHN

DEFENSE THREAT REDUCTION AGENCY
ALBUQUERQUE OPERATIONS
1680 TEXAS ST. SE
KIRTLAND AFB, NM 87117-5669
   ATTN: CPT-D G BALADI
   ATTN: CPTO
   ATTN: NSID

DIRECTOR
DISA
TECHNICAL RESOURCES CENTER
7010 DEFENSE PENTAGON
WASHINGTON, DC 20301-7010
   ATTN: JNGO

DEPARTMENT OF DEFENSE CONTRACTORS

AEROSPACE CORP
P O BOX 92957
LOS ANGELES, CA 90009
   ATTN: D SCHMUNK M8-220
   ATTN: K G HOLDEN
   ATTN: LEE MENDOZA/M4090
   ATTN: N SRAMEK
   ATTN: R KOGA

ALLIED-SIGNAL, INC.
COMMUNICATIONS DIVISION
1300 EAST JOPPA ROAD
BALTIMORE, MD. 21286
   ATTN: DOCUMENT CONTROL

ANALYTIC SERVICES, INC. (ANSER)
1215 JEFFERSON DAVIS HIGHWAY
SUITE 800
ARLINGTON, VA 22202-2898
   ATTN: A SHOSTAK

BOEING CO
P O BOX 3707
SEATTLE, WA 98124-2207
   ATTN: M/S 2T-50 D EGELKROUT

BOEING NORTH AMERICAN INC.
AUTONETICS & MISSILES SYSTEMS DIVISION
P O BOX 3105
ANAHEIM, CA 92803-3105
   ATTN: V DE MARTINO

BOOZ ALLEN & HAMILTON INC
8283 GREENSBORO DRIVE
MCLEAN, VA 22102
   ATTN: D VINCENT
   ATTN: L ALBRIGHT

CALIFORNIA INSTITUTE OF TECHNOLOGY
JET PROPULSION LAB
4800 OAK GROVE DRIVE
PASADENA, CA 91109
   ATTN: C BARNES MS-303/220

CHARLES STARK DRAPER LAB, INC.
555 TECHNOLOGY SQUARE
CAMBRIDGE, MA 02139
   ATTN: J BOYLE, MS6E

CLEMSON UNIVERSITY
PHYSICS & ASTRONOMY DEPT
CLEMSON, SC 29634-1911
   ATTN: P J MCNULTY

COMPUTER PRODUCTS A DIVISION OF AMPEX
DATA SYSTEMS CORP
P O BOX 999
EL SEGUNDO, CA 90245
   ATTN: K WRIGHT

DEFENSE GROUP, INC
P O BOX 7522
SANTA MONICA, CA 90406-7522
   ATTN: ROBERT POLL

EATON CORP.
AIL DIVISION
COMMACK ROAD
DEER PARK, NY 11729
    ATTN: R BRYANT
 FULL CIRCLE RESEARCH INC
SAN MARCOS, CA 92069
P O BOX 4010   ATTN: J SPRATT

GENERAL ELECTRIC CO.
DEFENSE SYSTEMS DEPARTMENT
100 PLASTICS AVENUE
PITTSFIELD, MA 01201-3698
    ATTN: B FLAHERTY, RM 2563
    ATTN: L HAUGE

GEORGE WASHINGTON UNIVERSITY
SCHOOL OF ENG & APPLIED SCIENCES
2121 I STREET NW
WASHINGTON, DC 20052
    ATTN: A FRIEDMAN

HARRIS CORPORATION
ELECTRONIC SYSTEMS SECTOR
P O BOX 37
MELBOURNE, FL 32902
    ATTN: E YOST
    ATTN: W ABARE, MS 101/4825

HONEYWELL INC
SOLID STATE ELECTRONICS CENTER
12001 STATE HIGHWAY 55
PLYMOUTH, MN 55441
    ATTN: C SANDSTROM M/S 4025

HONEYWELL, INC.
SPACE SYSTEMS DIVISION
SPACE AND STRATEGIC SYSTEMS OPR
13350 U.S. HIGHWAY 19 NORTH
CLEARWATER, FL 34624-7290
    ATTN: MS 725-5

HUGHES AIRCRAFT COMPANY
SPACE AND COMMUNICATION GROUP
P O BOX 92919
LOS ANGELES, CA 90009
    ATTN: E KUBO/S4/X301

IBIS TECHNOLOGY CORP
32-A CHERRY HILL DRIVE
DANVERS, MA 01923
    ATTN: MICHAEL L ALLES

IBM CORP.
THOMAS J WATSON RSCH CTR
P O BOX 218
YORKTOWN HEIGHTS, NY 10598-0218
    ATTN: A SADANA

ITT INDUSTRIES
ITT SYSTEMS CORPORATION
ATTN: AODTRA/DASIAC
1680 TEXAS ST SE
KIRTLAND AFB, NM 87117-5669
    ATTN: DASIAC
    ATTN: DASIAC/DARE
    ATTN: R RUTHERFORD

JAYCOR
1410 SPRING HILL ROAD
SUITE 300
MCLEAN, VA 22102
    ATTN: DR CYRUS P KNOWLES

JAYCOR
25 NORTH CASCADE AVENUE
COLORADO SPRINGS, CO 80903-1644
    ATTN: JERRY I LUBELL

JAYCOR
P O BOX 85154
SAN DIEGO, CA 92186-5154
    ATTN: D WALTERS

JOHNS HOPKINS UNIVERSITY (THE)
APPLIED PHYSICS LAB
11100 JOHNS HOPKINS ROAD
LAUREL, MD 20723-6099
    ATTN: R MAURER

KEARFOTT GUIDANCE AND NAVIGATION CORP.
150 TOTOEA ROAD
WAYNE, NJ 07470
    ATTN: J D BRINKMAN, MC 12B74

LITTON SYSTEMS INC
GUIDANCE & CONTROL SYSTEMS DIVISION
5500 CANOGA AVENUE
WOODLAND HILLS, CA 91367
    ATTN: F MOTTER

LOCKHEED MARTIN CORPORATION
P O BOX 3504
SUNNYVALE, CA 94088-3504
    ATTN: J CAYOT, 0/L4-30, BLDG 157
    ATTN: ORG 81-40, BLDG 157, G LUM

LOCKHEED MARTIN CORPORATION
SANDERS, A LOCKHEED MARTIN CO
P O BOX 2057
NASHUA, NH 03061-0868
    ATTN: BRIAN G CARRIGG, MS NCA1-6236

LOCKHEED MARTIN FEDERAL SYSTEMS, INC
9500 GODWIN DRIVE
MANASSAS, VA 22110-4198
   ATTN: L ROCKETT

LOCKHEED MARTIN FEDERAL SYSTEMS, INC
9500 GODWIN DRIVE
MANASSAS, VA 22110-4198
   ATTN: MURTY S. POLAVARAPU
   ATTN: N HADDAD

LOCKHEED MARTIN VOUGHT SYSTEMS
P O BOX 650003
DALLAS, TX 75265-0003
   ATTN: LIBRARY EM-08.

LOGICON RDA
2100 WASHINGTON BLVD
ARLINGTON, VA 22204-5706
   ATTN: DAVE CARLSON

MARTIN MARIETTA
CONTROL SYSTEMS
600 MAIN STREET
JOHNSON CITY, NY 13790-1888
   ATTN: J MILLER

MARTIN MARIETTA DENVER AEROSPACE
ASTRONAUTICS GROUP
P O BOX 179
DENVER, CO 80201
   ATTN: RESEARCH LIBRARY

MARYLAND, UNIVERSITY OF
DEPARTMENT OF ELECTRICAL ENGINEERING
COLLEGE PARK, MD 20742
   ATTN: H C LIN

MAXWELL TECHNOLOGIES CO
MAXWELL FEDERAL DIVISION INC
8888 BALBOA AVENUE
SAN DIEGO, CA 92123
   ATTN: M GERSTEN
   ATTN: MR RAUNCH
   ATTN: N LOTER
   ATTN: W RIX

MAXWELL FEDERAL DIVISION, INC
ALBUQUERQUE DIVISION
2501 YALE BLVD, S.E.,
SUITE 300
ALBUQUERQUE, NM 87106
   ATTN: DR JASON WILKENFELD

MISSION RESEARCH CORP.
1720 RANDOLPH ROAD, SE
ALBUQUERQUE, NM 87106-4245
   ATTN: D ALEXANDER
   ATTN: DAVID SLEETER
   ATTN: H JAKE TAUSCH

MITRE CORPORATION
202 BURLINGTON ROAD
BEDFORD, MA 01730-1420
   ATTN: J R SPURRIER
   ATTN: M FITZGERALD

ORBITAL SCIENCE CORP.
20301 CENTURY BLVD
GERMANTOW, MD 20874
   ATTN: ROB CHERNEY

PACIFIC-SIERRA RESEARCH CORP.
2901 28TH STREET
SANTA MONICA, CA 90405-2938
   ATTN: H BRODE

RAYTHEON SYSTEMS COMPANY
528 BOSTON POST ROAD
SUDBURY, MA 01776-3375
   ATTN: BERNARD HARRIS, MS 5-2-546
   ATTN: D D LEE
   ATTN: JOSEPH SURRO

RESEARCH TRIANGLE INSTITUTE
ISO
P O BOX 14086
RESEARCH TRIANGLE PARK, NC 27709-4086
   ATTN: M SIMONS

SARNOFF CORPORATION
CN 5300
ATTN: SECURITY
PRINCETON, NJ 08543-5300
   ATTN: R SMELTZER

SCIENCE APPLICATIONS INTL CORP
P O BOX 1303
MCLEAN, VA 22102
   ATTN: W CHADSEY

SCIENTIFIC RESEARCH ASSOC, INC.
P O BOX 498
GLASTONBURY, CT 06033-0498
   ATTN: H GRUBIN

SUNDSTRAND CORP.
P O BOX 7002
4751 HARRISON AVENUE
ROCKFORD, IL 61125-7002
   ATTN: C WHITE 745 M6

TELEDYNE BROWN ENGINEERING
P O BOX 070007
HUNTSVILLE, AL 35807-7007
   ATTN: G R EZELL/MS 19
   ATTN: M P FRENCH

TRW INC.
SPACE & ELECTRONICS GROUP
ONE SPACE PARK
REDONDO BEACH, CA 90278-1078
    ATTN: T.I.C., S/1930

TRW OGDEN ENGINEERING OPERATIONS
1104 COUNTRY HILLS DR.
OGDEN, UT 84403
    ATTN: WAYNE WAMPLER

TRW S. I. G.
STRATEGIC SYSTEMS DIVISION
P O BOX 1310
SAN BERNARDINO, CA 92402-1310
    ATTN: C BLASNEK/B 953/RM 1310

TRW SPACE & DEFENSE SECTOR SPACE &
TECH GROUP OGDEN ENG OPERATIONS
P O BOX 804
CLEARFIELD, UT 84015-0804
    ATTN: D M LAYTON
    ATTN: OUT6/HL DEPT/LIBRARY
    ATTN: OUT6/W WAMPLER

VISIDYNE, INC.
P O BOX 1399
GOLETA, CA 93116-1399
    ATTN: C H HUMPHREY
    ATTN: W P REIDY

WHITTAKER CORP
SAFETY SYSTEMS DIVISION
1915 VOYAGER AVENUE
SIMI VALLEY, CA 93063-3349
    ATTN: SECURITY OFFICER, MAIL CODE PER/97-
1399

DEPARTMENT OF ENERGY

DEPARTMENT OF ENERGY
(ALBUQUERQUE OPERATIONS OFFICE)
P O BOX 5400
ALBUQUERQUE, NM 87185-5400
    ATTN: NESD

UNIVERSITY OF CALIFORNIA
LAWRENCE LIVERMORE NATIONAL LAB
P O BOX 808
LIVERMORE, CA 94551-9900
    ATTN: L-183 G POMYKAL
    ATTN: L-303 W ORVIS

LOS ALAMOS NATIONAL LABORATORY
P O BOX 1663
LOS ALAMOS, NM 87545
    ATTN: MS B230/E LEONARD

SANDIA NATIONAL LABORATORIES
ATTN: MAIL SERVICES
P O BOX 5800
ALBUQUERQUE, NM 87185-0459
    ATTN: F SEXTON/MS 0527
    ATTN: L D POSEY/MS 1167
    ATTN: P WIN0KUR/MS 1083
    ATTN: T A DELLIN/MS 1071

DEPARTMENT OF THE AIR FORCE

AIR FORCE FOR STUDIES & ANALYSIS
1570 AIR FORCE PENTAGON
WASHINGTON, DC 20330-1570
    ATTN: SATI, RM 1D363

AIR FORCE RESEARCH LABORATORY
3550 ABERDEEN AVE., S.E.
KIRTLAND AFB, NM 87117-5776
    ATTN: PL/WSC

AIR UNIVERSITY LIBRARY
600 CHENNAULT CIRCLE
BLDG 1405 - ROOM 160
MAXWELL AFB, AL 36112-6424
    ATTN: AUL-LSE

ROME LABORATORY/CC
26 ELECTRONICS PARKWAY
GRIFFISS AFB, NY 13441
    ATTN: ESR

SMC/XR
155 DISCOVERER BL SUITE 2514B
LOS ANGELES AFB CA 90245-4692
    ATTN: MCX/LT DAN PHILLIPS

HEADQUARTERS
SPACE & MISSILE CENTER
LOS ANGELES AFB, CA 90090-5000
    ATTN: SMSC/MBP, KAREN BASANY

USAF ROME LABORATORY TECHNICAL LIBRARY,
FL2810
CORRIDOR W. STE 262, RL/SUL
26 ELECTRONICS PKWY, BLDG 106
GRIFFIS AFB NY 13441-4514
    ATTN: RBR

COMMANDER
WRIGHT LABORATORY/MTE
WRIGHT PATTERSON AFB, OH 45433
    ATTN: MTE

DEPARTMENT OF THE ARMY

US ARMY RESEARCH LABORATORIES
2800 POWDER MILL ROAD
ADELPHI, MD 20783-1197
   ATTN: AMSRL-WT-NJ
   ATTN: DR TIM OLDHAM

CHIEF
US ARMY RESEARCH OFFICE
P O BOX 12211
TRIANGLE PARK, NC 27709-2211
   ATTN: R GRIFFITH

COMMANDER
US ARMY SPACE & STRATEGIC DEFENSE
COMMAND
CSSD-TC-SR
P O BOX 1500
HUNTSVILLE, AL 35807-3801
   ATTN: CSSD-TC-SR

US ARMY THAAD PROJECT OFFICE
P O BOX 1500
HUNTSVILLE, AL 35807-3801
   ATTN: CSSD-WD

US ARMY WHITE SANDS MISSILE RANGE
DIRECTORATE FOR APPLIED TECHNOLOGY,
TEST AND SIMULATION
STEWS-DT-AC
WHITE SANDS MISSILE RANGE, NM 88002-5158
   ATTN: ALFREDO RAMIREZ

COMMANDER
USAISC
FORT HUACHUCA, AZ 85613-5000
  ATTN: ASOP-DO-TL BLDG 61801

DEPARTMENT OF THE NAVY

COMMANDER
NAVAL AIR SYSTEMS COMMAND
47123 BUSE RD, #IPT
PATUXENT RIVER, MD 20670
   ATTN: AIR-536T

NAVAL RESEARCH LABORATORY
4555 OVERLOOK AVE, SW
WASHINGTON, DC 20375-5000
   ATTN: CODE 6612 D BROWN
   ATTN: CODE 6613 A B CAMPBELL
   ATTN: CODE 6813 N SAKS
   ATTN: CODE 6816 H HUGHES

COMMANDING OFFICER
NAVAL WEAPONS SUPPORT CENTER
CRANE, IN 47522
   ATTN: CODE 6054 D PLATTETER, BLDG 2088

OFFICE OF NAVAL INTELLIGENCE
4251 SUITLAND ROAD
WASHINGTON, DC 20395-5720
   ATTN: LIBRARY

OTHER GOVERNMENT

CENTRAL INTELLIGENCE AGENCY
WASHINGTON, DC 20505
   ATTN: OSWR/STD/MTB 5S09 NHB

NASA
GODDARD SPACE FLIGHT CENTER
GREENBELT, MD 20771
   ATTN: C MARSHALL
   ATTN: K LABEL